UMinho | 2020

César Augusto Lourenço Cachulo

Children Dyscalculia: A Web System for Diagnosis and Treatment

César Augusto Lourenço Cachulo

Children Dyscalculia
A Web System for Diagnosis and Treatment

January 2020

Universidade do Minho
Escola de Engenharia

César Augusto Lourenço Cachulo

Children Dyscalculia
A Web System for Diagnosis and Treatment

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de

Professor Victor Alves
Mestre Filipa Ferraz

Janeiro de 2020

Acknowledgments

Thank you to everyone who played a role in my academic path up until now and that allowed the conclusion of this project.

To my parents, a special thank you for unwaveringly supporting me academically and emotionally since 11th September, 2001, my first school day, to the day that marks the completion of my master's degree.

To my little brother, a thank you for helping the completion of this project and for being available for any need that there could have been.

To my supervisors, professor Victor Alves and engineer Filipa Ferraz, for being readily available to solve any problem or doubt that existed, and for ensuring that the correct procedures were being followed for this project.

Thank you all!

STATEMENT OF INTEGRITY

Título: Discalculia em crianças: Sistema *Web* para diagnóstico e tratamento

Resumo

A discalculia é uma desordem neurológica que dificulta a aprendizagem de diversos conceitos matemáticos. Sendo uma desordem inerente à pessoa, pois nasce com ela, é crucial dedicar-lhe atenção desde cedo. Uma forma de atenuar os seus efeitos é o uso de tecnologia moderna como aplicações móveis, nomeadamente jogos com foco na prática e ensino de diversos conceitos relacionados com matemática, nos primeiros anos de vida escolar em pessoas que sofrem de discalculia.

Este projeto tem como foco a criação de uma plataforma cujo objetivo é complementar estes jogos registando diversos parâmetros relacionados com o desempenho dos alunos em diversas tarefas e retirando conclusões baseadas nestes parâmetros de forma a detetar a discalculia e os seus sintomas, assim como acompanhar o progresso do aluno no que toca ao tratamento desta desordem.

A criação desta plataforma permitiria a existência de uma nova e útil ferramenta no que toca à discalculia e ao tratamento dos seus sintomas o mais cedo possível na vida de uma pessoa.

Palavras-chave: Discalculia, crianças, aplicação móvel, jogo, plataforma *web,* base de dados, interface de programação de aplicações, aplicação *web*, análise de dados, modelação

Title: Children Dyscalculia: A Web System for Diagnosis and Treatment

Abstract

Dyscalculia is a neurological disorder that hampers the ability to learn certain mathematical concepts. It is a disorder that exists in the person since the moment of birth and, as such, it should be dealt with as early as possible. It is possible to mitigate the damage done by this disorder by using modern technology, particularly mobile games which are focused on developing abilities in various fields of mathematics, on the first years of children's scholar education.

The main objective of this project is the creation of a platform that is meant to complement these games by registering performance parameters that were achieved by the students and drawing conclusions based on said parameters in order to detect dyscalculia and its symptoms, as well as register the student's progress when it comes to its treatment.

By creating this platform there is a new and helpful tool regarding dyscalculia and the treatment of its effects, as early as possible in a person's life.

Keywords: Dyscalculia, children, mobile application, game, web platform, database, application programming interface, web application, data analytics, modeling

Table of contents

List of symbols and abbreviations

API	Application Programming Interface
DBMS	Database Management System
GUI	Graphical User Interface
JSON	JavaScript Object Notation
RDBMS	Relational Database Management System
SQL	Structured Query Language
SWOT	Strengths, weaknesses, opportunities and threats
UML	Unified Modeling Language

List of figures

List of tables

Glossary

Application Programming Interface	System that serves as an interface between different software programs in order to make software implementation or usage simpler
Class diagram	UML modeling language diagram that represents the entities that exist in a project or system, their attributes, the functions they do and the way they are connected to each other
Cloud database	Database hosted on a cloud computing platform. It is usually a paid service that comes together with a collection of other services that assure its maintenance and proper functioning
Command line	Computer program's command processor
Conceptual model	Representation of a system's components and concepts in a way that allow the comprehension of said system's purpose
Data analytics	Data analysis process that focuses on facilitating conclusion making about certain aspects of the data that is being analyzed
Data normalization	Process that aims to organize database data by reducing data duplication and by assuring data is in its proper place
Database	Data storage software system
Database query	Request regarding a database's data, be it data visualization or manipulation
Database table	Way of storing data within a database, representing an entity, it's attributes and their values
Developmental dyscalculia	Neurological disorder that hamper the learning and assimilation process of mathematics and arithmetic concepts
Foreign key (database)	Attribute or group of attributes that define relationships between database tables
Graphical User Interface (GUI)	Interface that allows a user to interact with a computer program
Java	Object oriented computer programming language
JavaScript	Object oriented computer programming language used mainly for web page and web application building
JavaScript Object Notation (JSON)	Data exchange format between computer systems
Logical model	Representation of a system or project in such a way that makes it understandable

	how it is going to be implemented without going into detail how it will be done
Machine learning technology	Technology that focuses on the automatic learning process by machines. It can be understood as artificial intelligence
Mockup	Graphical model of a project or system
Neuroplasticity	The brain's ability to change during a person's life in order to make it work more effectively
Physical model	Representation of a system or project in a way that displays how said system is or is going to be implemented
Primary key (database)	Unique attribute or group of attributes that define an entity in a database
Relational Database Management System	Software system that allows the user to change a database's data and details
Route	Part of the address of a web page or web application used for navigating them
Structured Query Language (SQL)	Database data management language
TypeScript	Object oriented computer programming language used mainly for web page and web application building that expands upon the capabilities of the JavaScript language
Unified Modeling Language	Software project modeling language
Use case diagram	UML modeling language diagram that represents a system or project from a user's perspective by showing the functions and actions said user can do
Virtual machine	Emulation of a computer system
Web application	Software program that can be accessed by using a web browser software
Web framework	Software designed for web application development
SWOT analysis	Planning procedure aimed towards identifying a project's strengths, weaknesses, opportunities and threats

1. Introduction

Dyscalculia is a neurological disorder characterized that renders the learning process of mathematics and arithmetic difficult. It has a lot in common with the better known disorder dyslexia, recognized for hindering another set of skills, reading and writing. The main difference lies in the fact that while dyslexia creates problems related with letters dyscalculia does the same but with numbers, mathematics signs and perception like time, distance and direction, among others [1]–[3].

One of the main concerns about dyscalculia is the fact that is not possible to be completely treated due to the fact that it is a neurological disorder, however its effects can be reduced to a point where they are not apparent. The most effective way to do this is during the first years of life of an individual when the brain is still developing. This means that a child that is known to suffer from dyscalculia in the first of school is at the most effective age to hamper the effects of dyscalculia [1], [2].

Nowadays it is possible to mitigate the effects caused by dyscalculia by making use of modern technologies, namely mobile applications such as games. A learning process directed towards children that relies on these tools is a powerful aid against this disorder and helps said children develop their abilities in the field of mathematics [4].

If there is a platform that complements these games, namely by registering scores in several tasks and the time taken to solve them, it is possible to detect signs of dyscalculia as well as the area of mathematics where the disorder is most prevalent and create a powerful aid in the fight against dyscalculia.

2. Dyscalculia

Dyscalculia is a neurological disorder that hampers the learning process of arithmetic concepts. Its origin can be traced back to a visual perception deficit or sequential orientation problems just like the better-known disorder dyslexia, which is responsible for difficulties related with reading and writing. As such, the concept of dyscalculia is used as the definition of the inability or difficulty to perform arithmetic operations. It is commonly associated with dyslexia due to the high likelihood of both existing within the same individual but is not as well-known since the people who suffer from it have an average or above average IQ, one of the defining features of both disorders. The major difference lies in the difficulty to deal with mathematical concepts, symbols, directions and other arithmetical concepts [1], [3].

The following sections will present some important topics regarding dyscalculia and explain its most relevant details having in mind the most relevant information to make this project as effective and helpful as possible as a tool to fight dyscalculia.

2.1. Symptoms

Since dyscalculia is a neurological disease its treatment is not possible in its entirety although its effects can be hampered majorly but in order for it to be possible its detection is a decisive factor and if it is made in the early stages of life of an individual it can be even more impactful [1].

The symptoms exhibited by dyscalculia all involve mathematics or arithmetic based origins and, in order for it to be detected, problems of this nature must be noticed. These symptoms include:

- Confusion and problems recognizing and understanding mathematical operations symbols (addition, subtraction, multiplication and division);
- Problems reading numbers in the correct order, either by reversing or by completely changing the order of its digits;
- Difficulty reading numbers with multiple digits, especially if they contain zeros;
- Problems dealing with the concepts of weight, capacity and distance as well as converting between their units;
- Difficulty to understand the concept of time as well as estimate its passage;
- Poor sense of direction even with auxiliary tools such as maps and compasses;
- Difficulty in everyday tasks such as counting change and remembering phone numbers and postal codes [1], [5], [6].

2.2. Consequences

The main consequence of dyscalculia is the inability or difficulty to deal with any kind of mathematics and arithmetic problems. This can cause some disturbances in the life of an individual that suffers from this disorder since it makes it hard to deal with certain aspects of everyday life which require the use of skills related with these areas of knowledge. Due to these disturbances it is not uncommon for an individual to develop disinterest and even aversion to mathematics since it can be very hard to deal with several concepts related to mathematics and even simpler aspects of life like deciding which side is left and which side is right [1].

But even though these consequences of dyscalculia are problematic not all individuals who suffer from it have difficulty in the same areas of knowledge as others. Some people have more difficulty dealing with arithmetic operations, others have trouble dealing with measures and converting between their units and other can even have problems dealing with their own sense of direction among others. As such are different types of dyscalculia in order to identify where the areas of difficulty of an individual reside:

- Lexical dyscalculia – associated with reading mathematical symbols;
- Verbal dyscalculia – associated with quantities, numbers and symbols;
- Graphic dyscalculia – associated with writing mathematical symbols;
- Operational dyscalculia – associated with mathematical operations;
- Practognostic dyscalculia – associated with enumeration, manipulation and comparison of objects;
- Ideagnostic dyscalculia – associated with the understanding of mathematical concepts and mental operations [1], [5].

2.3. Treatment

Dyscalculia is a disorder that can have its origins on different factors such as brain injury, poor education at school or at home, short memory problems or it can be due to heredity. As such it is a disorder that cannot be completely treated but its effects can be reduced to a point where is not noticeable. The best way to make this possible is to try to mitigate its effects as soon as possible in a person's life especially during the first years of education when the brain is still developing and learning mathematics and arithmetic related concepts for the first time. This also make its detection extremely a very important factor when dealing with it [1], [2], [6].

This disorder is due to a malformation on the brain that makes it hard to develop and understand certain concepts like perception, space, time and rhythm. There are some processes that were developed in order to modify the way the brain works so that it is possible to make the assimilation of concepts related to mathematics and arithmetic, namely the manipulation of neuroplasticity [1].

Neuroplasticity is the ability the brain has to change throughout a person's life with the main objective of making it work more efficiently and better. These changes can be triggered by behavior, environment, thoughts and emotions and they also can impact learning and memory skills among others, being important in the recovery process of brain damage. Many parts of the brain can be altered by this ability even during adulthood but it is during the developing phase of the brain that this is more recurring making it the reason why it is during the early stages of life that dyscalculia must be fought [7].

As such, the treatment of dyscalculia goes through a lot of brain, mathematics and arithmetic exercises that develop the part of the brain responsible for the learning deficit. In order to make this process more efficient some measure can be taken both at home and at school:

- Allow the use of paper and even finger counting when counting and making arithmetic operations;
- Make drawings of math problems in order to make them easier to understand;
- At school allow extra time or completely remove the time limit to solve tasks and even tests, as well as provide a quiet workplace so it is easier for the child to concentrate;
- Make available some helpful tools when solving math problems such as easy to use calculators, pencils and erasers;
- Make use of computer or mobile math games since it is possible to practice more time than with an educator and captive the attention of a child in an easier fashion [2], [6], [8].

3. Technologies and methods

The construction of this project was divided into three main components: the database, which stores all collected data, the web application itself and an application programming interface that connects the other two components (Figure 1).

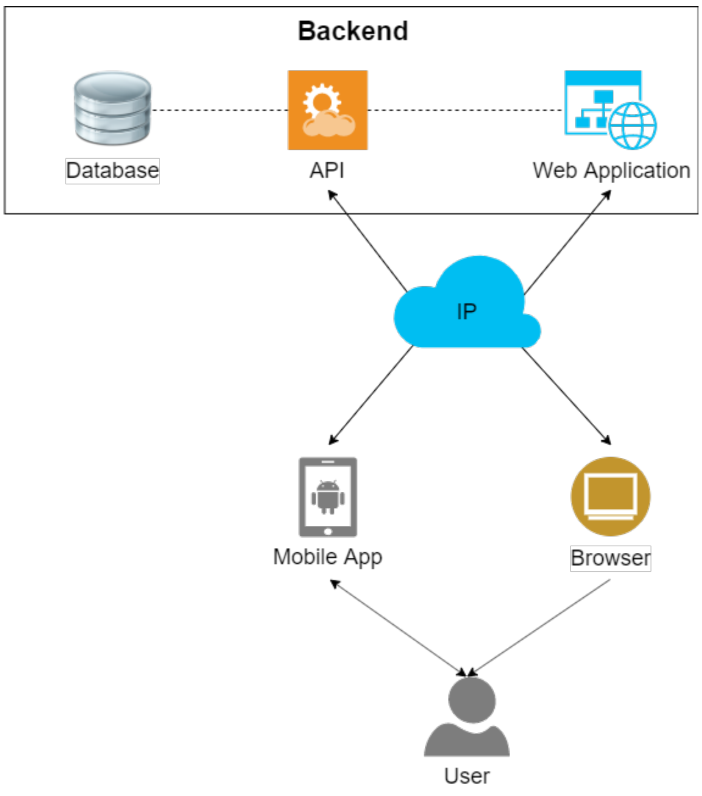


Figure 1 - Project components

The main objective of the database is to store all the data that the web application uses. This data consists of information regarding students, teachers, experts and the mobile games that are registered with the application, although, for this project, there is only one game and the scores and times achieved in each task by the student.

The web application is the focus of this project as it is what will display the data existing in the database and make use of that data to detect dyscalculia within the students as well as the areas of mathematics where it is more prevalent.

The application programming interface is the component of the project that connects the web application and the database. Whenever the web application needs to do any kind of operation requiring the database, be it data access, addition, removal or alteration, it will connect to the application programming interface which will request said operation from the database and handle its response as necessary.

Bearing in mind these basic definitions of each component several technologies were considered, tested and chosen. This section contains information regarding every considered technology as well as the reasons behind each choice.

3.1. Database

The most important requirement that the database must comply to is the fact that it must be accessible from anywhere. As this requirement was of the upmost importance it was one of the first points of focus of this project when the database was being developed and two options were considered: either the database was hosted on a computer set for that purpose or it was a cloud database.

A cloud database is a database hosted on a cloud computing platform. One of the main advantages of this kind of service is the fact that the database provider maintains and assures the proper functioning of the database [9]. Other services might be provided such as database analytics, security and performance optimization among others.

The services that were considered regarding this option were Microsoft Azure SQL Database and Firebase Realtime Database based on previous experience, recommendations and research. These services were also well-known and very popular when taking into account the technologies that were going to be used.

A database hosted on a computer that is not the responsibility of any company or service provider has a major setback: a computer that is always running the database is required. Luckily, Professor Victor Alves is responsible for a computer that is always running available for this purpose and gently granted access to it, making this option a viable one.

Regarding this option, the only technology that was considered was Microsoft SQL Server due to past experiences and results. MySQL Server was thought of but it was not considered because it was similar to Microsoft SQL Server but there wasn't as much experience with it. No options based on a language that is not SQL were considered due to the extra resources that it would take to set them up, learn them and test them.

In the next chapters, the decision process of each technology is explained by analyzing their advantages and disadvantages and by comparing the main features of each one in the most comprehensible way possible.

3.1.1. Microsoft SQL Server

Microsoft SQL Server is a relational database management system, a software system that allows its users to define, create, maintain and control access to a relational database which is database based on the relational model, a method of structuring data based on relations. It was developed by Microsoft and released to the public in 1984 [10]–[13].

This software system is based on SQL, also known as Structured Query Language, which is a language with the main purpose of managing the data inside a relational database management system. SQL was developed during the 1970s at IBM by Donald Chamberlin and Raymond Boyce and it was first released in 1986 [14]–[16].

For the goal the project wants to achieve Microsoft SQL Server is a pretty solid choice since it is quite stable, fast and secure and with the help of SQL Server Management Studio it becomes considerably easier to use and manage, making the database building process simpler and swifter [17].

3.1.1.1. Microsoft SQL Server Management Studio

SQL Server Management Studio, also known as SSMS, is a software application developed by Microsoft that is used to configure and manage Microsoft SQL Servers and Microsoft SQL Databases. This software can also be used with both Microsoft SQL Server and Microsoft Azure SQL Cloud Database allowing easier access to its contents and properties [18].

SQL Server Management Studio’s most relevant functionalities include:

- Database and database object creation, modification and deletion;
- Query creation and execution which is useful for reading and writing data to a database as well as for other operations such as data filtering and database object size calculation among others;
- User creation and management as well as role attribution;
- The ability to back up a database’s data manually or on a schedule;
- The Object Explorer which allows the user to navigate all databases and manage all their objects and settings with more ease [19], [20].

3.1.2. Microsoft Azure SQL Database

Microsoft Azure is a cloud computing platform released in 2010 by Microsoft that provides over 600 cloud services such as virtual machines and file storage as well as mobile and web application support [21], [22].

One of those services is the Microsoft Azure SQL Database that together with Microsoft SQL Server Management Studio, discussed in the previous section, provides its users with the ability to access and manage a SQL database hosted on a cloud platform making it safer, faster and available everywhere [23], [24].

Microsoft Azure SQL Database's most relevant features include:

- Data protection by using encryption and user authentication as well user access control;
- Adaptive performance tuning that improves database performance by using machine learning technology;
- High performance database scaling allowing workload and resource optimization [24], [25].

3.1.3. Firebase Realtime Database

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011. As of 2014 it belong to Google [26].

It provides its users with 18 services and tools aimed towards high quality application building, user base growth and profit increase. One of the services that is designed for application building is the Firebase Realtime Database [26], [27].

The Firebase Realtime Database is cloud-hosted database where all data is stored as a JSON object and synced in real-time, allowing all users to access it's up to date data at any time [27], [28].

Firebase Realtime Database's most relevant features include:

- Realtime data synchronization which allows all users to see up to date data at any time;
- Easily accessible since it's possible to access it from a mobile device or web browser;
- Authentication tools that aids in user authentication;
- Security rules that determine who can read and write data in the database [28], [29].

3.1.4. Technology selection

As it was previously mentioned, the choice involving SQL Server required a computer that must always be running the database, a problem that was resolved thanks to Professor Victor Alves that kindly granted access to a computer that is available for that purpose.

Even though this problem was solved, it was first decided that the database would be a cloud database in order to avoid the extra workload on Professor Victor Alves' computer.

As such, a choice had to be made between Microsoft Azure SQL Database and Firebase Realtime Database, so their advantages and disadvantages were compared.

Some of Microsoft Azure SQL Database's main advantages consist of:

- High availability meaning multiple users can access the database's data at any time;
- Strong security focus which is important as the number of threats increase;
- User access control that determine what certain users can do within each server and database;
- High performance database scaling making resources available as needed in order to optimize resource usage;
- Data loss prevention measures such as multiple backups on different physical servers;
- Cost effective since the services must only be paid as they are needed.

On the other hand, some of its disadvantages are:

- Platform expertise is required in order to ensure high database performance and efficiency;
- Management such as server monitoring is needed to make sure everything is working as pretended [30], [31].

Firebase Realtime Database's most relevant advantages are:

- Easy access at any time from any mobile device or web browser;
- Authentication and security tools that only allow users to access and write data they are meant to.

Some of its disadvantages include:

- Very limited reporting and analytics tools;
- Querying is also very limited compared to other databases;
- Although it starts free it is also very limited and upgrading it can be very costly [32]–[35].

After analyzing all advantages and testing both platforms, Microsoft Azure SQL Database was firstly chosen. It was more advantageous than Firebase Realtime Database, having some very appealing features as the use of an already well-known language, SQL.

At first the database was created and maintained using Microsoft Azure SQL Database but during development there were some problems with the free trial that Microsoft offers, namely billing problems, and, as such, SQL Server running on Professor Victor Alves' computer was the preferred choice. The database was migrated to the new server and no other problems and difficulties arose which made SQL Server the final choice.

3.2. Web application

The web application is the focus of the project as mentioned previously since it is the component that will display and make use of the data that exists in the database to detect students that suffer from dyscalculia as well as the areas where the disorder is more prevalent.

As such, the web application must have an appealing and easy to understand interface. It also must have simple, yet complete, ways to show data in order to understand and make it easier to access the degree of dyscalculia the students might suffer from.

In order to build the web application itself it was decided that the tool that was going to be used was Angular. No other tool was considered due to previous experience and achieved results, which were very satisfactory, especially considering that the existing experience in any kind of web application building was quite small.

Regarding the data component of the web application there were two tools that were considered, researched and tested: Microsoft's Power BI and ElasticSearch. These were taken into account based on recommendations.

In the next sections the decision process between Microsoft's Power BI and ElasticSearch is explained by comparing features, advantages and disadvantages of each tool as well as explaining and discussing Angular and every auxiliary tool that was needed for the development of the web application itself.

3.2.1. Data analytics

Data analytics is defined as the process in which data is analyzed so that it can be possible to draw conclusions about the information it contains. It can be used in several areas such as finance, business, healthcare and security for example, in order to save money, increase efficiency of certain services and other ends [36], [37].

This project needs tools that are capable of performing data analytics in order to make it easier to detect dyscalculia and which areas of mathematics and arithmetic it is more prevalent within the students.

The most important feature these tools must have for the project is the ability to create and present graphs and other visual representations of data that can be used on the web application in a way that is aesthetically pleasing, simple and easy to understand and draw conclusions from.
As it was mentioned previously, the tools that were considered were Microsoft's Power BI and ElasticSearch, all based on recommendations. In the next chapters these tools are compared by analyzing their advantages, their disadvantages and their most important features.

3.2.1.1. Power BI

Power BI is a business analytics service developed by Microsoft that allows its users to create reports, a way to present data and draw conclusions from it in an easy to understand format [38], and dashboards, which organize and display data from multiple sources compactly [39].
It provides cloud based business intelligence services which are known as Power BI services as well as a desktop based interface called Power BI Desktop which primary functions are the design and publishing of data reports [40].
In 2016 Microsoft released an additional service called Power BI Embedded on its Azure cloud platform that makes it easier for its users to enrich their web applications by embedding analytics on them even if those applications are already designed [40], [41].
Power BI's most relevant features include:

- Easy creation of data rich interactive reports and dashboards;
- Data unification across multiple sources including SQL Servers, Excel, text files and other options;
- Several options regarding data representation and visualization [42].

3.2.1.2. ElasticSearch

ElasticSearch is an open source search engine developed by Shay Banon released in 2010. The main purpose of this tool is to store documents and allow them to be searched and retrieved by creating a searchable reference to each one.
Although ElasticSearch alone is not what is required for the data analytics component of the project, there is a plugin for it that allows the creation of interactive dashboards, which organize and display data from multiple sources compactly, as mentioned in the previous section, a feature that is in line with what is needed [43], [44].

Kibana is a free open source data visualization plugin for ElasticSearch that allows users to create histograms, line charts and heat maps among others and build interactive dashboards as well as embed them on their web applications and websites [45], [46].

ElasticSearch and Kibana's most relevant features include:

- Makes use of ElasticSearch search engine's capabilities in order to create data driven graphs and dashboards;
- High performance thanks to ElasticSearch's data searching capabilities which makes it possible to process large volumes of data and obtain the intended data quickly;
- Dashboard customizability with modern and clean aesthetics [44], [47], [48].

3.2.1.3. Technology selection

In order to make the decision between Power BI and ElasticSearch their advantages and disadvantages must be studied and compared.

Power BI's most relevant advantages to achieve the pretended objectives are:

- Familiar and intuitive user interface since its structure is similar to other Microsoft product such as the Microsoft Office's programs;
- Easy installation since all that is required is an internet connection and an updated computer that runs Microsoft Windows operative system to download it from the Microsoft Store;
- It allows the usage of multiple data sources, such as SQL databases, Excel spreadsheets and text files among others;
- Data refresh capabilities that allow all created graphs to have up to date data;
- Minimal to no cost since it is a subscription-based system with multiple plans that adapts to the user's needs.

Some of its most relevant disadvantages are as follows:

- Inability to deal with massive amounts of data;
- Although all the graphics are aesthetically pleasing they lack some configurability and customizability [49]–[53].

ElasticSearch most relevant advantages consist of the following:

- Free tool with a lot of options for data visualization;
- Data anomalies detection thanks to unsupervised machine learning technology;
- Aesthetically pleasing with graph customization capabilities;

On the other hand, some of its disadvantages are:

- Need to index SQL databases in order to take advantage of ElasticSearch and Kibana;
- Hard to use with a very steep learning curve;
- Prior knowledge of Java, JSON, search engines and web technologies required [47], [54]–[57].

Other than the better graph customizability options there seems to be no other major advantage of using ElasticSearch with Kibana. Its steep learning curve and the need to index our SQL Database also make it a little unappealing.

On the other hand, Power BI's user interface and easy data importing capabilities seem interesting and useful features. None of its disadvantages seemed to really be negative and damage the overall project development.

So, with all of this taken into account and after testing both options, the chosen tool was Microsoft's Power BI. Although, during the testing phases, Microsoft's Power BI fulfilled the needs that were initially informally planned for the web application, it was, in a later stage of the development process, deemed not the best or the most practical alternative, a situation that was also found to be true for ElasticSearch, since all graphs that were to be used had to pre-generated and stored in some way,

As such, during the development of the web application, some difficulties arose due its lack of practicality. So, before trying to find a software program that better suited the needs of the project, two libraries of functions and components that exist for the chosen web application development tool, mentioned in the next section, were found to be what was needed when used together, with an aesthetically pleasing graphical presentation of its components, with some customization functions, and with the practicality that was required. Those libraries are known as Chart.js and Ng2 Charts.

Charts.js' functions are used to set the data that is to be displayed in the graph, or graphs, that are to be shown in a page or web application. Ng2 Charts contains other functions mainly used to set customization options to any graph like colors or to set any label, and its colors, and others.

Although, when both libraries, Chart.js and Ng2 Charts, were being tested, they were being firstly used as a temporary alternative to a data analytics software so other aspects of the database could be developed. However, due to pleasant results, they were regarded as the final decision concerning the data analytics software technology to be used in the project.

3.2.2. Web application building: Angular

As previously mentioned, the web application's most important requirement is that it must have an appealing and easy to understand interface so that it is easier to retrieve the information needed in order to understand the degree of dyscalculia the students might suffer from in the simplest way possible.
With this requirement taken into account the chosen tool to build the web application was Angular. This was the only tool that was considered due to past experiences and very satisfactory results that were achieved while having little to no experience in web application building.
In the following sections Angular and its most relevant features regarding the web application building process of this project will be discussed, as well as every other tool that was required in order to use every pertinent capability it has in the best way possible and make the first drafts of what would become the visual component of the web application.

3.2.2.1. Angular

Angular is an open-source platform developed by Google focused on web application building. It is based on TypeScript that is a programming language developed by Microsoft that allows developers to write JavaScript code, a scripting language, which is a programming language based on the execution of tasks, that makes it possible to create interactive web pages [58]–[62].
The main requirement for the web application is that it must have and appealing and easy to understand interface, as previously mentioned. Thanks to an active community there are several libraries that offer a lot of aesthetically pleasing options regarding web application design, making Angular a good option for the design component of the project [63].
Angular's most relevant features include:

- Modern web platform capabilities that allow developers to build high performance web applications;
- Fast to start building, add components, test and deploy web applications;
- Component based architecture making it easy to replace a component if it does meet any requirement;
- Great offer of community developed libraries that provide several functionalities and components that allow developers to make their web applications fashionable, modern looking and very functional [63]–[65].

3.2.2.2. Visual Studio Code

Visual Studio Code is a text editor focused on programming code editing developed by Microsoft that supports JavaScript, TypeScript and Node.js. As mentioned on the previous section, JavaScript and TypeScript are programming languages that makes it possible to create interactive web pages. Node.js is an open-source tool created by Ryan Dahl that allows its users to run JavaScript code. It is quite useful for the project since it make it possible to test the web application, making Visual Studio Code a useful tool for the development of the project [66]–[69].

Some of its most relevant features include:

- JavaScript, TypeScript and Node.js support that allows developers to execute and test any kind of JavaScript files with little to no previous configuration;
- Integrated command line interface which makes it easier to test and manage the project;
- Repository with plugins that extend the code editing capabilities of Visual Studio Code by adding support for additional programming languages and other quality of life improvements [67], [70], [71].

3.2.2.3. Pencil

Pencil is a free open-source software that has the main objective of allowing its users to create mockups in an easy to set up and use fashion. A mockup is a draft that shows what a product, in the case of this project a web application, will look like or tend to look like [72], [73].

In order to start building the web application a mockup of it was required. Its main purpose is to make the building process more efficient. Since a web application takes time to build, testing several designs in order to achieve the desired look for it would be quite a time-consuming task. Therefore, a mockup should be created in order to test several designs and help visualize the intended look for the web application whilst saving a lot of time [74].

The most relevant features of Pencil include:

- Easy to start building mockups due to its simple interface and available tools;
- Several types of shapes are available such as general-purpose shapes like arrows, ovals and circles and web interface shapes such as buttons, tables and several bars;
- Community created shapes are available on the internet, most of them for free, with a very simple installation process;
- Easy to output any mockup to different types of picture format [75].

3.3. Application programming interface

The main objective of the application programming interface is to connect the web application to the database in order to perform any kind of data related operation in the simplest way possible, facilitating the integration of data in the application as well as its building process.

As such, the application programming interface should be efficient and fast in order to obtain the required data with the greatest ease possible. As such, the chosen tool for developing this component of the project was ASP.NET. This was the only considered option as it was a tool that was used in the past for a few other personal projects with satisfactory results and due to the fact that understanding how to build an application programming interface with any tool has a learning curve that requires a considerable amount of time to overcome.

In these next sections ASP.NET's most important features will be discussed as well as every other tool that was needed in order to be possible to build and host the application programming interface for the project.

3.3.1. ASP.NET

ASP.NET is a server-side tool developed by Microsoft which main objective consists of creating dynamic web sites and applications and web services. These web services include the ability to create application programming interfaces, the component that is required for the development of the project [76]–[78].

ASP.NET's most important and relevant feature for the project is its high performance which is due to the use of modern technologies that make it one of the fastest options available, something that is required as mentioned in the previous section [79].

3.3.2. Microsoft Visual Studio

Microsoft Visual Studio is a programming interface developed by Microsoft that makes it easier for developers to build software. It offers the ability to edit programming code as well as build and run developed code [80], [81].

This tool can be used to develop several kinds of software aimed at different ends since it supports a varied range of technologies and programming languages, including C#, the programming language used for the development process of the application programming interface, and ASP.NET, also developed by Microsoft [80].

The most important feature it has is its built-in support for ASP.NET. Since both tools are developed and maintained by Microsoft, Visual Studio was the natural choice to use in order to develop the application programming interface.

3.4. Hosting on the web: Internet Information Services

In order to make the project available from anywhere it had to be possible to access and use each of its components from anywhere as well. As discussed in the previous sections, these components are the database, the web application and the application programming interface.

The database uses Microsoft SQL Server and is running in a computer professor Victor Alves has available for the project. As for the web application and the application programming interface the computer is also available but in order to make them available on the web a tool was chosen to host them. This tool is Microsoft's Internet Information Services. The two main reasons why it was chosen were its simplicity and its ability to host both the web application and the application programming interface.

Internet Information Services is a web server tool developed by Microsoft. It is integrated with almost every Microsoft Windows operative system but it must be activated in order to be used since it's deactivated by default. Its main objective consists of hosting web related pages and files [82], [83].

Just like with Visual Studio's case, both ASP.NET and Internet Information Services were developed by Microsoft which also make Internet Information Services the natural choice for hosting the application programming interface component of the project.

Its most relevant features are its ability to host HTML pages as well as ASP.NET web applications and its ease of use since its very simple to host any pages and applications and make changes to them without dealing with complex setups or configurations.

4. Database building process

Having selected all the tools that were going to be used for building the project the next step had to be the construction of one of the components: the database, the application programming interface or the web application.

The decision that was taken was to build the database first due to a couple of reasons: its planning and construction would lead to a better understanding and overview of the overall project and what it would need and the application programming interface and the web application would need a working database in order to start developing their main features.

As mentioned in the previous section (Technologies and methods) Microsoft SQL Database was selected as the main tool for the database. Having done that selection the next step that was needed in order to build the database was to plan the building process so that it fulfills every requirement needed for the project. As such, some steps were necessary:

- Case study analysis – the analysis of the purpose and goals the database aimed to achieve;
- Requirements analysis – the listing of every requirement the database must fulfill;
- Requirements validation – the verification process of the list of requirements made in the previous step;
- Database design – the building process of the database model that is in accordance with the list of requirements made in the previous step;
- Database implementation – the implementation and deployment processes of the database that allow it to be publicly available;
- Database testing – the testing and correction process of potential problems the database might have [84]–[86].

4.1. Case study

Dyscalculia is a disorder that makes it harder to learn several mathematical concepts and like stated previously is a disorder that cannot be completely treated but its effects can be hampered to a point where they are not noticeable. One of the best resources to make this possible are mathematics related mobile games, a tool that is especially effective the younger the person who uses it is.

Having this in mind, this project focuses on the creation of a web application that manages and shows data collect from children playing said mobile games with the main objectives of detecting and diagnosing dyscalculia and help keep track and develop their abilities in the fields of the mathematics and arithmetic so that their deficiencies are not as hampering in their life.

As such, the database that the project needs is very important for its objectives since it will be the component needed to save and hold all the data that ensures its proper functioning.

4.2. Goals

Considering the nature of this project and that the purpose of databases is the storage and serving of data upon request, the database must fulfill certain objectives so that the project and the functionalities it promises can be achieved.

As such, these objectives are:

- Store information regarding all entities are involved with the web application;
- Store information regarding all the available games that were developed in conjunction with this project (as of the making of the project there is only one) as well as the most relevant information pertinent to them;
- Store the scores and times achieved by the students in said games as well as other important data.

These objectives were the starting point in the building process of the database and the first step towards listing and validating the requirements that were discussed in the next section.

4.3. Requirements analysis

Having settled which are the main goals the database must achieve the next step is to define the requirements necessary so that those goals can be met. As such the goals that were stated in the previous section will be discussed as well as the requirements it takes for them to be possible.

The first objective that was listed was the ability to store information regarding all entities involved with the web application. This means that the database must be able to store personal information and other functionality-based information, like names, identification codes and classes for example. The data that exists in the database must be so that the following requirements are met:

- There should be tables to store information regarding students, teachers, experts and developers/administrators;
- There should be a table in order to store information regarding classes, like the students it has and the teacher and expert that were assigned to them, for example.

The second objective that was stated was the ability to store information regarding all existing games developed having this platform in mind. This requires the possibility to save such data as the game's name, all its levels and tasks (a game contains levels that contain tasks, a structure that should be followed by any game that has this web application in mind), as well as the area or areas of dyscalculia that a certain task focuses on. As such, the requirements that the database must comply to regarding this topic are:

- There should be tables that are used to store information regarding an application, or mobile game, like its name and its identification code, for example;
- There should be tables similar to the previous ones but regarding the levels that exist in an application;
- There should also be tables that focus on the tasks that exist in a level of an application that can contain information such as the area of dyscalculia that said task is designed for.

The thirds objective was the ability to store scores and time achieved by the students in each task of each game. This implies that there should be data containing scores and the amount of time it took to complete a task as well as the date and time those same scores were achieved. So, the requirements for the third objective are as follows:

- There should be a table that will be used to store information regarding the score and time achieved by a student in a level of an application as well as the date and time of completion;
- There should be a table that stores information related to the previous table but regarding the score and time achieved by a student in a task of a level of an application.

Having made the listing of requirements the next step was to assure that they followed the goals and objectives this project strives to achieve.

4.4. Requirements validation

Regarding the first objective of the database, the requirements that were listed make it possible since the users that will make use of the web application and that are directly related to it can be represented and information can be stored about them.

Regarding the second objective, it is possible to store data about any application that can be integrated with the web application as well as its levels and tasks by using the tables that were described regarding said objective in the previous section.

Regarding the third objective of the database, it can be achieved since all the information that is needed for the web application can be stored by following the requirements that were presented in the previous section.

With all the requirements listed and validated, which makes it possible to build a database with all the functionalities the project promises, it was possible to try to make the model of what would become the database.

4.5. Conceptual model

The primary objective of this step was to build the conceptual model of the database, a model which main purpose is to show how a system is structured and how it can achieve what it is supposed to achieve in the simplest way possible. The most important characteristic of this type of model is the fact that it only represents the entities and relationships among them, their attributes and defining features are not shown in order to only represent the most important concepts [87]–[90].

As such, these entities and relationships should be defined, only then the conceptual model can be designed. The entities that were defined for the database component of this project are as follows:

- Student – the entity responsible for storing data related to the students associated with the platform;
- Teacher – the entity responsible for storing data related to the teachers associated with the platform;
- Expert – the entity responsible for storing data related to the experts associated with the platform;
- Developer – the entity responsible for storing data related to developers and administrators associated with the platform;
- Class – the entity responsible for storing data related to existing classes;
- Application – the entity responsible for storing data related to applications and mobile games associated with the platform;
- Application level – the entity responsible for storing data related to application levels;
- Application task – the entity responsible for storing data related to application tasks;
- Student score on application level – the entity responsible for storing data related to the scores and time achieved by students on a level of an application;
- Student score on application task – the entity responsible for storing data related to the scores and time achieved by students on a task of an application.

A relationship between two or more entities represent the way they work together, and an important fact is that some entities depend on others so that they can exist. As such, the relationships that were defined for this step of the database building process are:

- Class and student – a relationship which main objective is to understand which students belong to which class;
- Class and teacher – a relationship that aims to represent the teacher that is responsible for a class;
- Class and expert – a relationship that aims to show the experts that is responsible for a class;
- Application and application level – a relationship that exists in order to show the levels that exist in an application;
- Application level and application task – a relationship that aims to show the tasks that exist within a level of an application;
- Student and student score on application level – a relationship that exists to make it possible to understand which student achieved a score on a level of an application;
- Student score on application level and application level – a relationship that aims to represent the level of an application that a level score refers to;
- Student score on application task and student score on application level – a relationship which main objective is to make it understandable to which level's score an application task is related to;
- Student score on application task and application task – a relationship that aims to show the task of an application level a task score refers to.

After those components were defined it was possible to build the conceptual model of the database. This was done by using a software called Visual Paradigm. Visual Paradigm is a computer program build with software and project design as its main objective (Figure 2). It offers support for some modeling languages, including UML, also known as Unified Modeling Language, the one that was decided it was going to be used due to past experience with it [91].

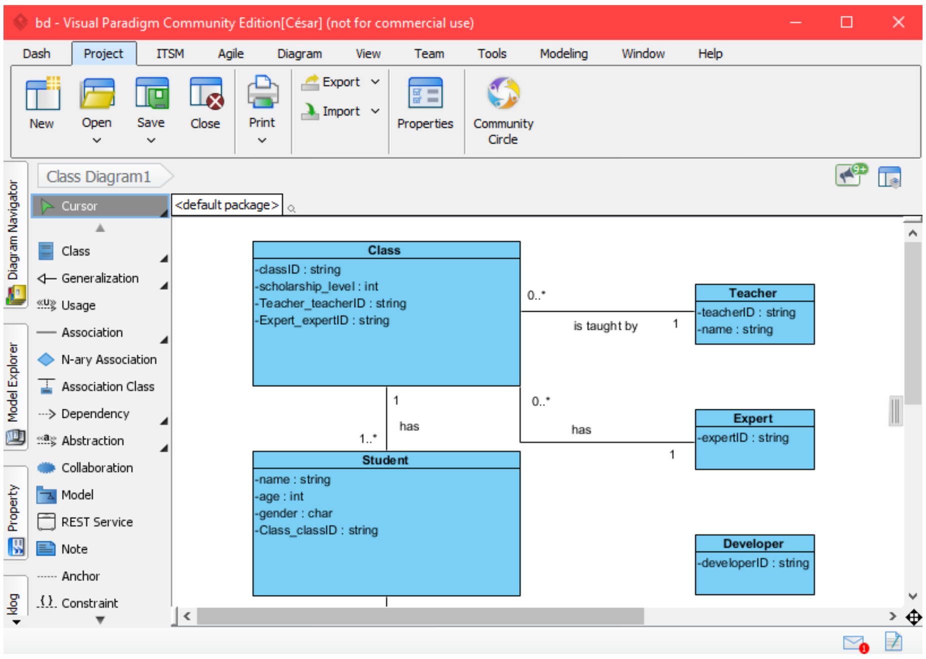


Figure 2 - Visual Paradigm software preview

The chosen type of diagram for representing the database was a class diagram, a type of diagram belonging to the UML modeling language that focuses on representing the classes that exist in a project or system, their attributes, the way they are connected with each other and even the functions they are responsible for regarding the system, although, as mentioned at the start of this section, only the classes and the relationships between them will be represented [92].

In summary, the result of this step is a class diagram that represents the database this project is expected to need in such a fashion that shows all the previously defined entities as well as the relationships that were established between them (Figure 3).

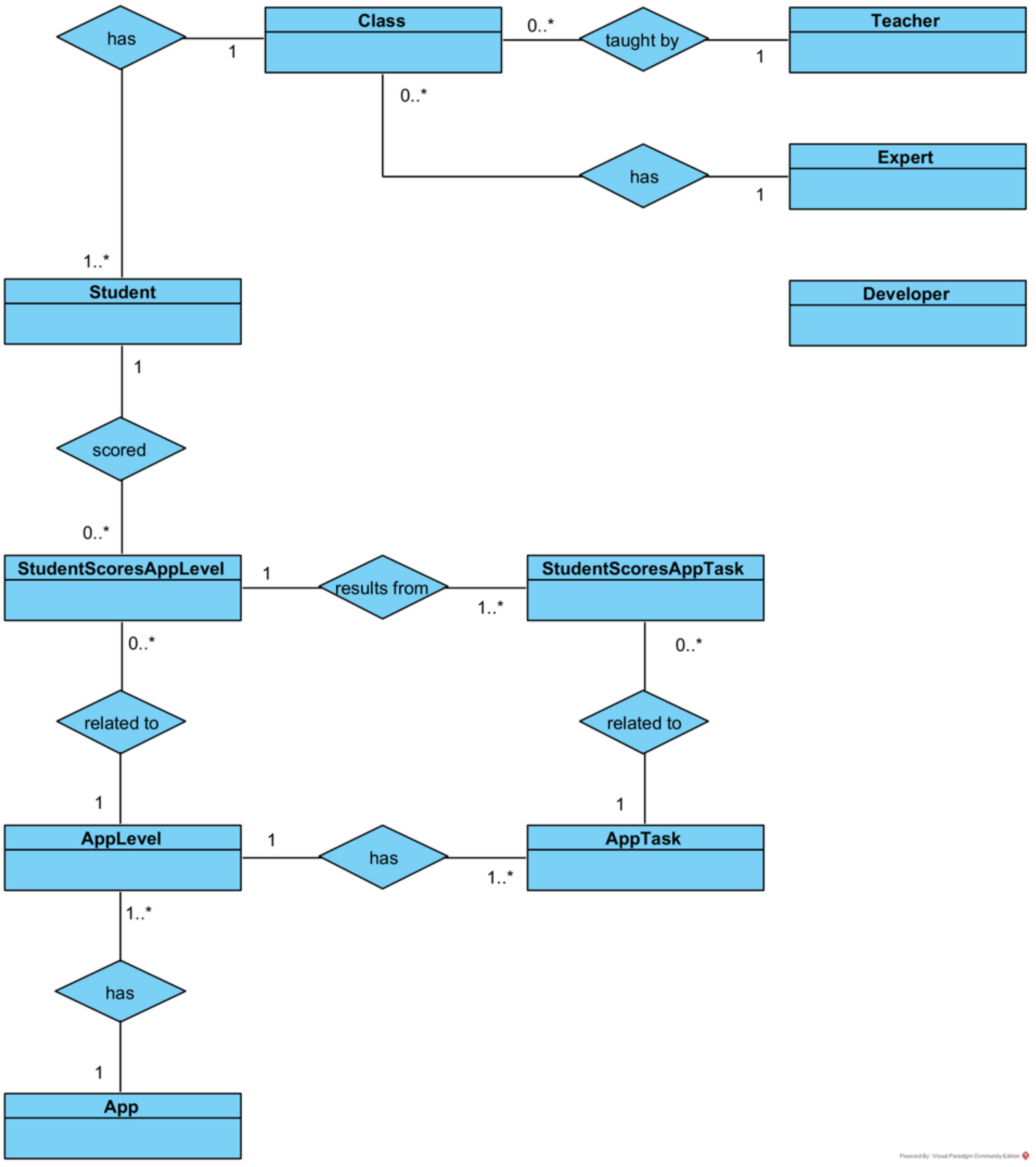


Figure 3 - Database conceptual mode

On this point of the development process of the database, all the necessary conditions were met to start building a more complex model that will serve as the basis to build the database itself, the logical model.

4.6. Logical model

The following step towards building the database was the design of the logical model, a more complex model based on the previously conceived conceptual model.

A logical model is a representation of a system or project in such a way that it describes the data that will be implemented without going detailing how it will be physically implemented in the final product or result. Just like conceptual models it will represent entities and relationships but it will expand their definition by adding attributes and defining keys [89], [90].

The attributes of a class or entity are the features that detail them further, for example, if there was an entity called person some of its attributes would be its name, age and gender.

Regarding keys, there are primary keys and foreign keys. A primary key is the attribute or attributes that define an entity and each primary key is unique, for example, is there was an entity called person and its primary key would be its national identification number there could not be two people with that same number since there would be no way to differentiate between several instances of the same entity [90].

A foreign key an attribute or group of attributes that define relationships between entities. For example, if there were two entities, person and vehicle, the vehicle could have an attribute called owner which was the national identification number of its owner, the primary key of the entity person, defined as example in the previous paragraph. The attribute owner in the vehicle entity would be a foreign key since it establishes a relationship between two entities [90].

Since the conceptual model was defined, it was used as the starting point to start building the logical model. This was done by following some steps:

- Defining primary keys for the existing entities;
- Defining relationships between entities as well as foreign keys;
- Defining the attributes for every entity;
- Resolving possible many-to-many relationships;
- Normalizing the data [90].

In order to define primary keys every entity was analyzed in order to ensure that every primary key was in accordance with the needs of the database. The results of this process, for each entity, are as follows:

- Student – an identification code called *studentID;*
- Teacher – an identification code called *studentID;*
- Expert – an identification code called *expertID;*
- Developer – an identification code called *developverID;*
- Class – an identification code called *classID;*

- Application – an identification code called *applicationID;*
- Application level – a key composed by multiple attributes, the identification code of the application it is related to and the number of the level;
- Application task – a key composed by multiple attributes, the identification code of the application and the number of the level it is related to and the number of the task;
- Student score on application level – a key composed by multiple attributes, the identification codes of the student and application, as well as the number of the level it is related to, and the date (including time) in which the score and time were achieved;
- Student score on application task – a key composed by multiple attributes, the identification codes of the student and application, as well as the numbers of the level and task it is related to, and the date (including time) in which the score and time of the level score was registered.

With the primary keys settled, the next step was the definition of relationships and foreign keys. This was done with the existing relationships of the conceptual model in mind. As such, the resulting foreign keys for each relationship were the following:

- Class and student – the identification code of the class the student belongs to is added to the student table, an attribute called *Class_classID;*
- Class and teacher – the identification code of the teacher responsible for the class is added to the class table, an attribute called *Teacher_teacherID;*
- Class and expert – the identification code of the expert responsible for the class is added to the class table, an attribute called *Expert_expertID;*
- Application and application level – the identification code of the application is added to the application level table, an attribute called *App_appID;*
- Application level and application task – the identification code of the application and the number of the level are added to the application task table. These attributes are called *App_appID* and *AppLevel_levelNumber;*
- Student and student score on application level – the student's identification code is added to the level score table, an attribute called *Student_studentID;*
- Student score on application level and application level – the application's identification code and the level number the score refers to is added to the level score table. These attributes are called *App_appID* and *AppLevel_levelNumber;*
- Student score on application task and student score on application level – the identification code of the student and application as well as the level number and the date in which the scores were

achieved are added to the task score table. There attributes are called *Student_studentID*, *App_appID*, *AppLevel_levelNumber* and *StudentScoresAppLevel_scoreDate;*

- Student score on application task and application task – the application's identification code as well as the level and task numbers are added to the task score table. The attributes are called *App_appID*, *AppLevel_levelNumber* and *AppTask_taskNumber*.

Having decided the keys for every entity and relationship, deciding the attributes for each entity was next in the line work regarding the construction of the logical model. The process was reminiscent of the process done for defining the primary keys since each attribute should be well thought of so that the database is in accordance with its requirements. That being, the attributes defined for each entity are as follows:

- Student – the primary key *studentID*, the attributes *studentPassword*, *studentName*, *studentNumber*, *studentAge*, *studentGender* and the foreign key *Class_classID;*
- Teacher – the primary key *teacherID* and the attributes *teacherPassword* and *teacherName;*
- Expert – the primary key *expertID* and the attributes *expertPassword* and *expertsName;*
- Developer – the primary key *developerID* and the attributes *developerPassword* and *developerName;*
- Class – the primary key *classID*, the attributes *scholarshipLevel*, *schoolName* and the foreign keys *Teacher_teacherID* and *Expert_expertID;*
- Application – the primary key *appID* and the attribute *appName;*
- Application level – the foreign key *App_appID* which is part of the primary key alongside *levelNumber;*
- Application task – the foreign keys *App_appID* and *AppLevel_levelNumber*, which are part of the primary key alongside *taskNumber*, and the attribute *dyscalculiaArea;*
- Student score on application level – the foreign keys *Student_studentID*, *App_appID* and *AppLevel_levelNumber* which are part of the primary key alongside *scoreDate* and the attributes *levelScore* and *levelTime;*
- Student score on application task – the foreign keys *App_appID*, *AppLevel_levelNumber*, *AppTask_taskNumber* and *StudentScoresAppLevel_scoreDate*, which form the primary key of this table, and the attributes *taskScore* and *taskTime*.

The next step was the resolution of many-to-many relationships. Many-to-many relationships are relationships that are the product of multiple records of an entity being associated with multiple records of another entity (or the same as well). For example, in a library setting, a person can read multiple books and a book can be read by multiple people [93].

In order to resolve these kinds of relationships between two entities, a table that represent said relationships should be created. That table is connected to the entities tied by the many-to-many relationship and it contains the primary keys of both entities. Using the same example of the previous paragraph, in order to resolve the relationship between the entity person and the entity book a third table would be created. That table would contain the primary key for person, for example the national identification number, and the primary key for book, the library code for the book, for example [93].

Luckily, by analyzing the conceptual model, it was possible to establish that there were no existing many-to-many relationships, a fact that made this step a simpler one.

The final step towards building the logical model was the normalization of the data. Data normalization is a process that aims to reduce data redundancy, also known as duplicate data, and ensure that the proper data is stored in its respective table. According to the Theory of Data Normalization there are several stages regarding data normalization known as normal forms. As of the making of this project, there are six normal forms, although, for most applications, normalization is considered to be at its best on its third form [94]–[96].

The first normal form is achieved when the following conditions are met:

- The data is stored in tables in which each record is identified by a unique primary key;
- The data within each attribute is in its most reduced form;
- There are no repeating attributes [94], [96].

This means that every table should have a unique primary key, each attribute should be in its most reduced form, for example, if there was a table person with an attribute called *name_and_gender* the data would not be normalized and there should be two attributes to represent the original attribute, and there are no repeating attributes, for example, if the same table as the previous example, person, had three attributes called mobile_number_1, mobile_number_2 and mobile_number_3, the data would not be normalized and there should be another table called mobile, with the *mobile_number* attribute, so that there only was the necessary amount of data stored in the database [94], [96].

By analyzing the previously defined tables for the entities and their relationships as well as their attributes it is possible to conclude that every condition is met, and that the final product of this step should be complacent to the first normal form of data normalization.

The second normal form is achieved when these next requirements are met:

- The first normal form is met;
- All attributes are directly related to their respective entity's primary key [94], [96].

These requirements mean that every condition listed for the first normal form should be met and that the attributes in an entity's table should be related to that table, for example, if there was a table called dissertation with the attributes *dissertation_code*, *dissertation_title, dissertation_author_name* and *dissertation_author_mobile_number* that table would not be normalized since the last two attributes are not directly related to the dissertation entity itself and it should be in another table designed for authors with their name and mobile number [94].

Resulting from the examination of all attributes and tables, the second form is already achieved since the first normal form is met, as seen before, and there are no attributes that are not related to their respective table's nature.

In order to achieve normalization, there is still a normal form left, the third normal form. The third normal form requires the following requirements to be achieved:

- The conditions required for the second normal form to be achieved are met;
- All attributes in a table should only be dependent on the primary key and no other attribute [94]–[96].

This means that each listed requirement for the second normal form should be met, which means that the first normal form's requirements should also be met, and that all attributes should only be related to the primary key of their table, for example, if there was a table tournament with the attributes name, year, winner and *winner_date_of_birth* that table would not be normalized since the attribute *winner_data_of_birth* would be related to the attribute winner. There should be a new table called winner which contained the winner's name and data of birth in order for the first table to be normalized according to the third normal form [94], [97].

Analyzing all defined entities, their attributes and all relationships that were established earlier in this section it is possible to conclude that the model would be in accordance with the third normal form of data normalization since it already was correct according to the second normal form and all attributes are related to the table they are in and not related to any other attribute.

Since all steps of the construction of the logical model were done at this point, the next step was to build the model itself. Just like the conceptual model the software that was used to design said model was Visual Paradigm and the chosen type of diagram was the UML modeling language's class diagram since it allows the possibility to expand the conceptual model with the new data defined for this model.

As such, the resulting logical model (Figure 4) for the database of this project is as follows:

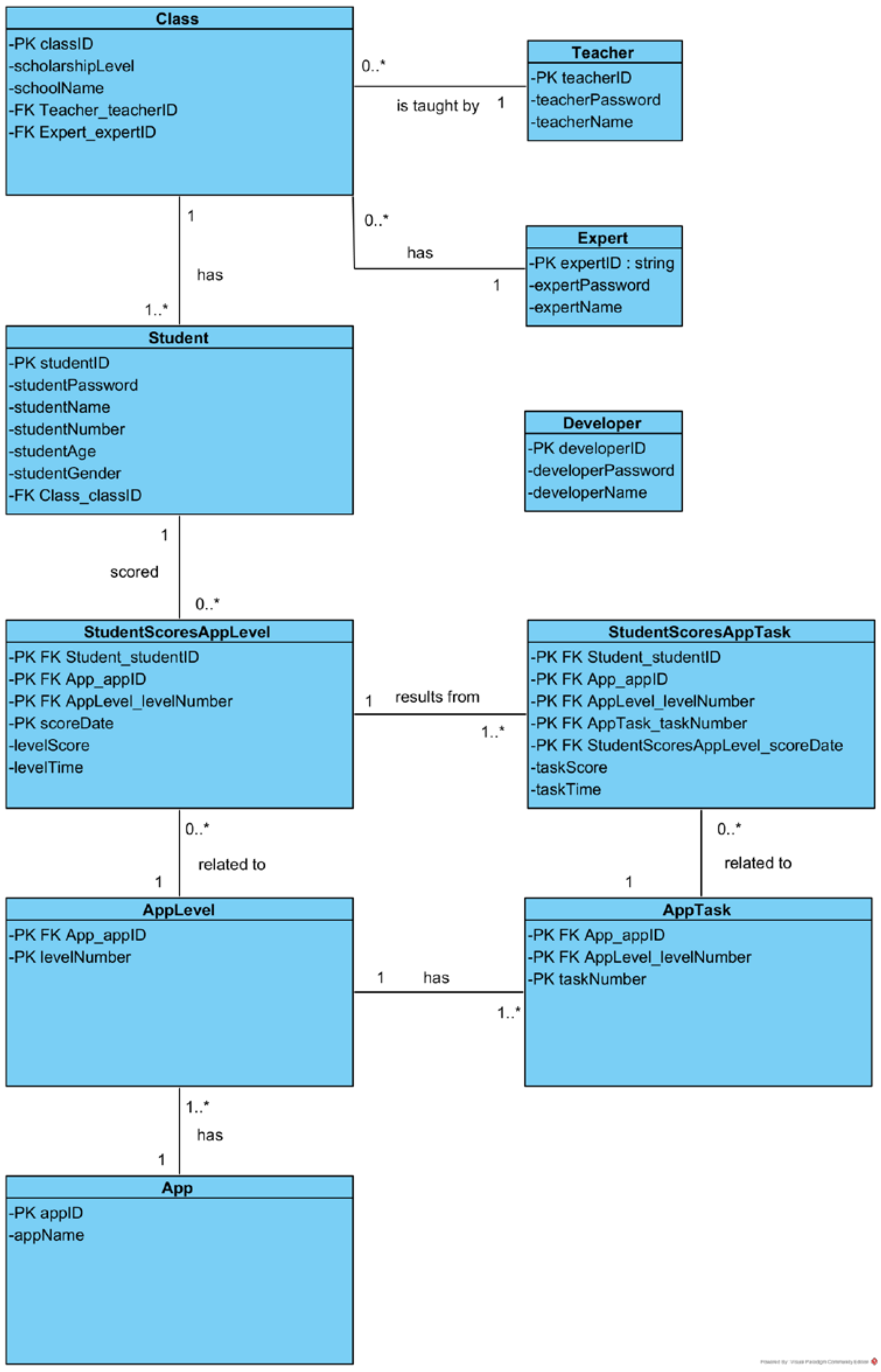


Figure 4 - Database logical model (PK – Primary key; FK – Foreign key)

The next step will result in an even more complex diagram that was used in order to produce the database that was planned and discussed in the last sections.

4.7. Physical model

The step discussed in this section shows the last model that was built and how the database itself was built based of said model.

The physical model is an even more complex model than the previous two that were conceived in the last sections, the conceptual model and the logical model, and represents how the database will be built. This model includes some key aspects:

- Tables representing the entities present in the database;
- Column names representing the entities' attributes;
- Column data types that represent the type of data that is going to be used for the respective attribute;
- Column constraints that represent certain rules or characteristics regarding the respective attribute such as NOT NULL, which states that an attribute can't be left blank, PRIMARY KEY and FOREIGN KEY, which represents an attribute that is a primary key or foreign key, respectively, and UNIQUE, which ensure that all records in a column are different;
- Primary and foreign keys;
- The relationships between the entities [89], [90], [98].

With this definition in mind the steps necessary in order to build the physical model starting from the logical model are as follows:

- Convert all entities and relationships into tables;
- Convert all primary and foreign keys into columns;
- Convert all attributes into columns;
- Set column data types and constraints based on the requirements of the database.

Since the logical model is a good representation of the database necessary for the project the first three steps were a simple task. The process that was done consisted in modeling what was designed in the logical model in the software program that was used to represent the physical mode, a program that is discussed further in this section. Because there was no new data to be added to the model at this point, the process surrounding the first three steps could be summarized to a software conversion process.

The last step is responsible for the biggest change between the logical model and the physical model since it adds new data, the data types assigned to each attribute and the constraints or rules said attribute must follow.

So, the results of this step are as follows, starting with the results related to the students table (Table 1):

Column name	Data type	Constraints
studentID	nvarchar(20)	PRIMARY KEY, NOT NULL
studentPassword	nvarchar(256)	NOT NULL
studentName	nvarchar(200)	NOT NULL
studentNumber	nvarchar(20)	NOT NULL
studentAge	int	NOT NULL
studentGender	nvarchar(1)	NOT NULL
Class_classID	nvarchar(20)	FOREIGN KEY, NOT NULL

Table 1 - Student physical model table

Before analyzing this table it is important to state that the data type column is used to represent the data format that will be used for the column it represents in the database software and the number that is between parenthesis represents its length. For example, this means that the *studentName* with the datatype nvarchar(200) is made up of characters and it can contain up to 200 characters.

When designing this table some decisions were made, decisions that would also transpose to other tables. Those decisions were that all identification codes were going to be of the type nvarchar(20), all names of the type nvarchar(200) and all passwords of the type nvarchar(256). One important detail to note is the password length. When thinking of this column it was decided that the password would be encrypted in order for it to be stored in the database. As such, its length is related to the encryption algorithm that was chosen, a process that will be discussed in another section.

The rest of the table is straightforward. The student number was not assigned a numeric data type because some school numbers include letters, the student gender length is only one due to it being represented by a single character, M for male and F for female, and the student age has a numeric data field.

More results are presented in the following tables (Table 2, Table 3 and Table 4):

Column name	Data type	Constraints
teacherID	nvarchar(20)	PRIMARY KEY, NOT NULL
teacherPassword	nvarchar(256)	NOT NULL
teacherName	nvarchar(200)	NOT NULL

Table 2 - Teacher physical model table

Column name	Data type	Constraints
expertID	nvarchar(20)	PRIMARY KEY, NOT NULL
expertPassword	nvarchar(256)	NOT NULL
expertName	nvarchar(200)	NOT NULL

Table 3 - Expert physical model table

Column name	Data type	Constraints
developerID	nvarchar(20)	PRIMARY KEY, NOT NULL
developerPassword	nvarchar(256)	NOT NULL
developerName	nvarchar(200)	NOT NULL

Table 4 - Developer physical model table

These tables are rather simple, especially due to the decision that was described in the previous table explanation were the identification code, name and password columns' data type were described.

Column name	Data type	Constraints
classID	nvarchar(20)	PRIMARY KEY, NOT NULL
scholarshipLevel	int	NOT NULL
schoolName	nvarchar(200)	NOT NULL
Teacher_teacherID	nvarchar(20)	FOREIGN KEY, NOT NULL
Expert_expertID	nvarchar(20)	FOREIGN KEY, NOT NULL

Table 5 - Class physical model table

The results presented in the previous table (Table 5) show the column related to the class table of the database. Most of its datatypes are straightforward thanks to the decision explained in the second paragraph of the student results' table (Table 1). Some details worth noting include the scholarship level field that is defined as a numeric field that represents the school year of the class, and the school name, a field that, even though it does not represent a person's name, is still a name and, as such, follows the previously decided format.

The following tables show the results that were obtained regarding the database tables related to the application (Table 6), application level (Table 7) and application task (Table 8):

Column name	Data type	Constraints
appID	nvarchar(20)	PRIMARY KEY, NOT NULL
appName	nvarchar(200)	NOT NULL

Table 6 - Application physical model table

Column name	Data type	Constraints
App_appID	nvarchar(20)	PRIMARY KEY, FOREIGN KEY, NOT NULL
levelNumber	int	PRIMARY KEY, NOT NULL

Table 7 - Application level physical model table

Column name	Data type	Constraints
App_appID	nvarchar(20)	PRIMARY KEY, FOREIGN KEY, NOT NULL
AppLevel_levelNumber	int	PRIMARY KEY, FOREIGN KEY, NOT NULL
taskNumber	int	PRIMARY KEY, NOT NULL
dyscalculiaArea	nvarchar(20)	NOT NULL

Table 8 - Application task physical model table

Almost all fields follow the previously defined format and, as such, their definition process was also straightforward. The level number and the task number columns are defined as a numeric filed and the dyscalculia area field of the application task is defined by a field containing up to 20 characters. This field must also follow a format that is discussed in a later section of the development process of this project.

The last results are related to the tables that contain the scores achieved by students on levels (Table 9) and tasks (Table 10):

Column name	Data type	Constraints
Student_studentID	nvarchar(20)	PRIMARY KEY, FOREIGN KEY, NOT NULL
App_appID	nvarchar(20)	PRIMARY KEY, FOREIGN KEY, NOT NULL
AppLevel_levelnumber	int	PRIMARY KEY, FOREIGN KEY, NOT NULL
scoreDate	datetime	PRIMARY KEY, NOT NULL
levelScore	int	NOT NULL
levelTime	int	NOT NULL

Table 9 - Student score on application level physical model table

Column name	Data type	Constraints
Student_studentID	nvarchar(20)	PRIMARY KEY, FOREIGN KEY, NOT NULL
App_appID	nvarchar(20)	PRIMARY KEY, FOREIGN KEY, NOT NULL
AppLevel_levelnumber	int	PRIMARY KEY, FOREIGN KEY, NOT NULL
AppTask_taskNumber	int	PRIMARY KEY, FOREIGN KEY, NOT NULL
StudentScoresAppLevel_scoreDate	datetime	PRIMARY KEY, NOT NULL
taskScore	int	NOT NULL
taskTime	int	NOT NULL

Table 10 - Student score on application task physical model table

Since these tables' columns are in their majority foreign keys their implementation was, once again, straightforward since it was equal to the one in the table they are taken from. The most notable details include the score date of both tables that represent the data in which the score of a level was registered, that follows a data format called datetime, and the score and time fields that are represented by numeric fields. It is important to note that the time field is represented in seconds.

After all steps were completed all the data needed to build the physical model was collected and all that was needed was to build the model itself. In order to help represent it, the logical model was updated with the new data that was defined.

The result (Figure 5) was the following model:

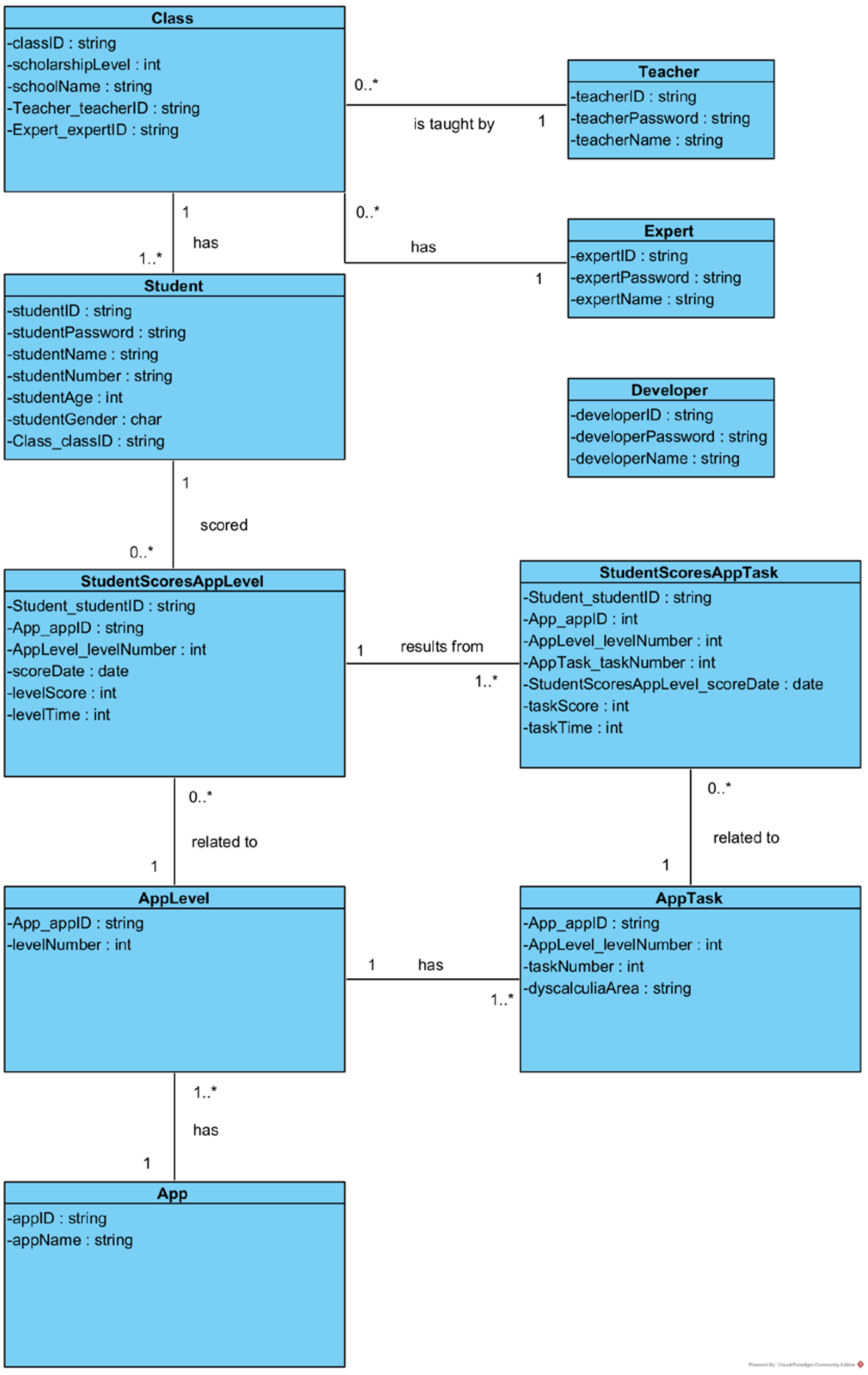
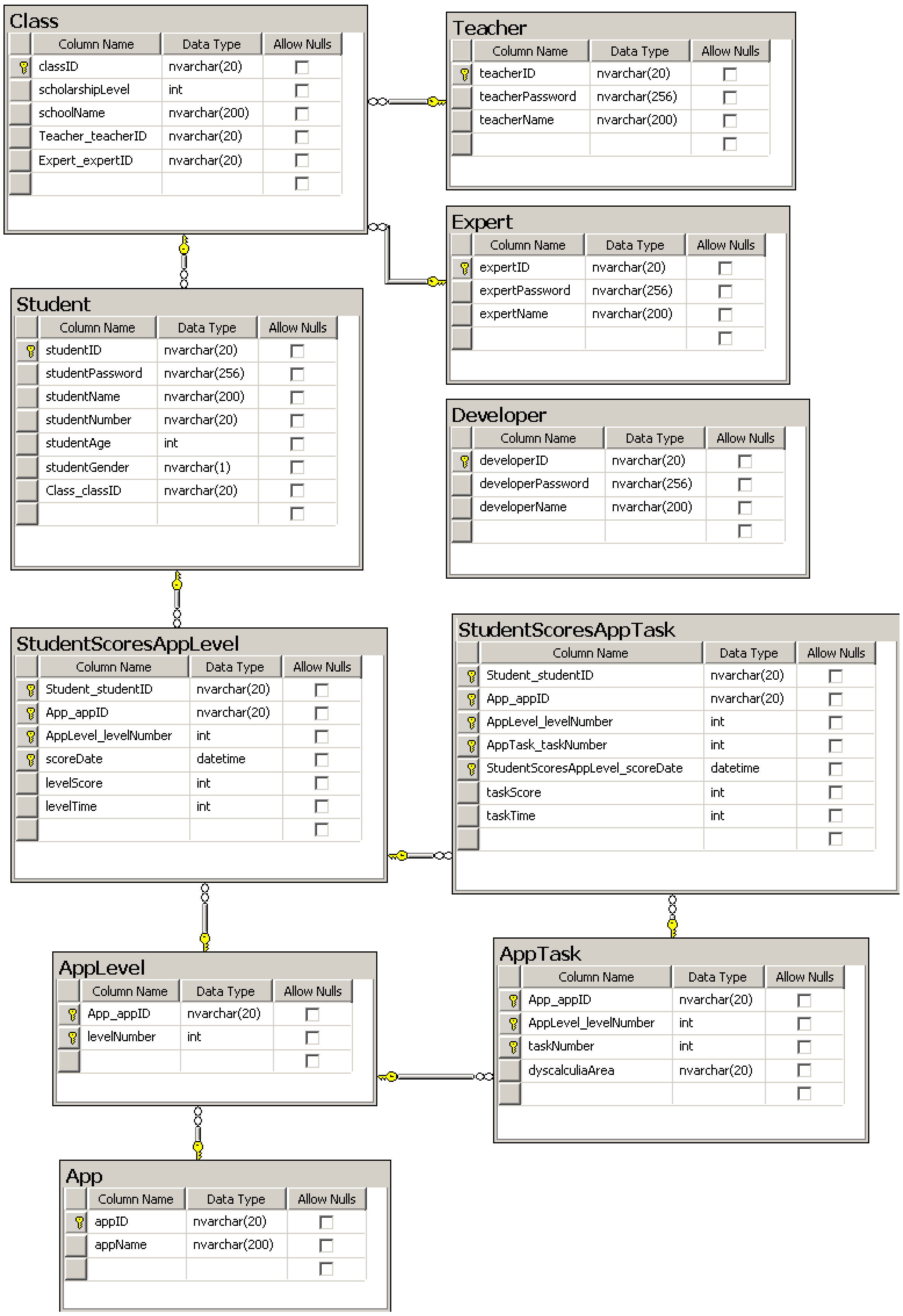


Figure 5 - Database physical model (PK – Primary key; FK – Foreign key)

With the physical model defined it was possible to build the database. The software that was decided to be used for the database was Microsoft SQL Server as mentioned in the Technologies and methods section. By using Microsoft SQL Server Management Studio, another software program mentioned in the Technologies and methods section, the database could be built on a previously set up Microsoft SQL Server running in a computer gracefully provided by professor Victor Alves for the purpose of this project. The construction of the database done by converting the physical model built using Visual Paradigm into the SQL format, a language designed to manage a database's data. Using the graphical interface capabilities of Microsoft SQL Server Management Studio this process required little to none typing and was quite fast since the model that the database was going to be based upon was already designed.

The achieved result (Figure 6) was as follows:



Class

Column Name	Data Type	Allow Nulls
classID	nvarchar(20)	☐
scholarshipLevel	int	☐
schoolName	nvarchar(200)	☐
Teacher_teacherID	nvarchar(20)	☐
Expert_expertID	nvarchar(20)	☐
		☐

Teacher

Column Name	Data Type	Allow Nulls
teacherID	nvarchar(20)	☐
teacherPassword	nvarchar(256)	☐
teacherName	nvarchar(200)	☐
		☐

Expert

Column Name	Data Type	Allow Nulls
expertID	nvarchar(20)	☐
expertPassword	nvarchar(256)	☐
expertName	nvarchar(200)	☐
		☐

Student

Column Name	Data Type	Allow Nulls
studentID	nvarchar(20)	☐
studentPassword	nvarchar(256)	☐
studentName	nvarchar(200)	☐
studentNumber	nvarchar(20)	☐
studentAge	int	☐
studentGender	nvarchar(1)	☐
Class_classID	nvarchar(20)	☐
		☐

Developer

Column Name	Data Type	Allow Nulls
developerID	nvarchar(20)	☐
developerPassword	nvarchar(256)	☐
developerName	nvarchar(200)	☐
		☐

StudentScoresAppLevel

Column Name	Data Type	Allow Nulls
Student_studentID	nvarchar(20)	☐
App_appID	nvarchar(20)	☐
AppLevel_levelNumber	int	☐
scoreDate	datetime	☐
levelScore	int	☐
levelTime	int	☐
		☐

StudentScoresAppTask

Column Name	Data Type	Allow Nulls
Student_studentID	nvarchar(20)	☐
App_appID	nvarchar(20)	☐
AppLevel_levelNumber	int	☐
AppTask_taskNumber	int	☐
StudentScoresAppLevel_scoreDate	datetime	☐
taskScore	int	☐
taskTime	int	☐
		☐

AppLevel

Column Name	Data Type	Allow Nulls
App_appID	nvarchar(20)	☐
levelNumber	int	☐
		☐

AppTask

Column Name	Data Type	Allow Nulls
App_appID	nvarchar(20)	☐
AppLevel_levelNumber	int	☐
taskNumber	int	☐
dyscalculiaArea	nvarchar(20)	☐
		☐

App

Column Name	Data Type	Allow Nulls
appID	nvarchar(20)	☐
appName	nvarchar(200)	☐
		☐

Figure 6 - Microsoft SQL Server database model

At this point of the development of this project, the only necessary step to have the database available was to deploy it. This process was done by recurring again to Microsoft SQL Server Management Studio and made it so that the database was ready to be accessible through a public address on the internet, allowing for the construction and development of the other components of this project.

4.8. Database testing

Having the database ready and deployed the bulk of the work regarding this component of the project was done. The possible remaining work regarding the database would be related to possible problems or new requirements it could have. As such, and to avoid any unnecessary and major remodeling, it had to be tested.

Database testing is a process that aims to ensure the integrity of the data in a database, its consistency and its responsiveness. This process is usually done by creating complex request to the database to test all of its components, which in the case of Microsoft SQL server, the database used for this project, include tables, constraints, relationships and other components not mentioned nor needed for the project such as procedures, SQL language snippets, and triggers, a component that trigger an action when certain conditions are met [99]–[101].

There are several software tools available for this purpose, but since the database for this project is not complex nor is expected to be submitted to a heavy traffic load, such tools are considered not to be necessary [99].

The initial testing done to the database was quite simple. It consisted of simple random manual insertion, removal and alteration of data in tables. This was done as a first test and a mean to assure that the database software was also working and running without problems.

The next stage of testing that was done to the database was a bit more demanding. Engineer Filipa Ferraz provided data that was achieved by collecting results achieved by Portuguese primary school students playing a mobile game project in which she was part and that could be used to test the database and the other components of this project as well.

In order to do that a SQL script, a file that contains one or more SQL commands, was developed to test the insertion of a considerable amount of data into the database in a small amount of time. That process was repeated a few times by removing the inserted data and running the script again and it never proved to be a problem regarding either the data insertion operations or the database response time.

The remaining tests that were done consisted of operations performed during the development and testing periods of the other components of this project, the application programing interface, which will be discussed in the next section, and the web application that is connected to the database by said application programming interface. During the development of this project there was a mobile game project directed towards dyscalculia that was also being developed at the same time. This game uses the application programming interface of this project and sends the data it collects to the database of this project. As such, during its development and testing phases it was also a very useful resource to test the database and the application programming interface.

During these testing procedures, the database never suffered any major change. This was mostly due to the planning the database went through and the fact that it does not contain any complex elements that a more demanding and bigger database could have.

Since the database was ready at this point of the project with all its important aspects and features covered the next step of the project was the construction of the application programming interface, a component of the project that is heavily dependent of the database and that will be discussed in the next sections.

5. Application programming interface

Having the database up and running the next step was to decide what the next component to build would be: either the application programming interface or the web application. The component that was selected was the application programming interface for two main reasons, it was the less arduous of the two and the web application could not be completed entirely without the application programming interface implemented and running.

The technology that was selected for building the application programming interface was Microsoft's ASP.Net and the one that was selected to host it in the web was Microsoft's Internet Information Services. As such, this section will focus on these two processes and will discuss the way in which they were planned and accomplished.

In order to start developing the application programming interface component, the general steps of its construction had to have been planned to have the fastest and most efficient building process as possible. Having the same steps as the database development process as a starting point for the planning phase the results were as follows:

- Case study analysis – the analysis of the purpose and goals that the application programming interface must achieve;
- Requirements analysis – the listing of every requirement the application programming interface must fulfill;
- Requirements validation – the verification process of the list of requirements made in the previous step;
- Application programming interface implementation – the building process of the application programming interface in accordance with the list of requirements made;
- Application programming interface hosting - the hosting process of the application programming interface in order for it to be accessible through a public address on the internet;
- Application programming interface testing – the testing and correction process of potential problems the application programming interface might have [84]–[86].

5.1. Case study

At this point in the development process of this project, its database component was developed and ready, deployed and accessible through a public address. This database makes it possible to store data necessary to develop the web application and all of its features although it is not all that is needed.

An application programming interface, also known as API, is a component of a project that serves as a channel of communication between another component, in the case of this project the web application, and a database in such a way that it is easier or simpler to perform some operations such as data retrieval or addition, to a certain extent, even to the most common user [102].
The application programming interface of this project is the component that connect the web application, as well as the mobile games developed with this application in mind, which collect students' scores and times, to the database. By sending a request to this component it is possible to access, add, remove or alter data by simply being in possession of its public address, a process simpler than connecting to the database, producing a request, sending it and processing the response from the database every time a new operation needs to be done.
That being, the application programming interface is a component which is very helpful regarding the web application building process, which was the longest and most complex one of all the components of this project, since it cuts down the time and amount of coding needed to develop some of its features.

5.2. Goals

According to its definition and purpose, an application programming should allow a user to perform operations on a database's data. In the context of this project, these operations should be in accordance to the needs of the web application which, even if they are relatively simple, they are very important in order to allow more complex functionalities to be possible.
As such, the goals the application programming interface must achieve are:

- Allow a user to visualize requested data present on the database, if allowed to do so;
- Allow a user to add requested data to the database, if allowed to do so;
- Allow a user to remove requested data present on the database, if allowed to so;
- Allow a user to alter requested data present on the database, if allowed to so.

These goals were the starting point for the development process of the application programming interface and were the basis for listing the requirements this component of the project must comply to as well as the validation of said requirements according to the needs of the project.

5.3. Requirements analysis

The primary objective of this step is to define all the requirements needed in order to build an application programming interface that complies with the objectives listed in the previous section. As such, every objective will be analyzed as well as every aspect and detail needed for it to be achievable.

The first objective the application programming interface must comply to is the ability that users must have to visualize data if certain conditions are met. These conditions are different according to the type of user which makes said request. So, according to this, every entity of the database that can access the application was analyzed and the following requirements were drawn:

- Student – the student is the primary focus of the project since the project revolves around him and his potential difficulties. As such, when a student accesses the platform, the student should have access to all the information about him, his teacher and expert, his class, all apps, tasks and levels as well as his and his class's score records;
- Teacher – the teacher is the entity responsible for the learning process of the student. As such, he should have access to all the information about him, all students, all teachers, all experts, all apps, tasks and levels as well as all student's score records;
- Expert – the expert is the entity responsible for supervising all the aspects related to the scores achieved by the students. As such, the expert should have access to all the information about him, all students, all teachers, all experts, all apps, tasks and levels as well as all student's score records;
- Developer – the developer is an entity that contains developers as well as administrators of the web application. The focus of this entity is to assure that the platform is working as intended and that all existing data is correct. As such, the developer should have access to all the information existing in the database including students, teachers, experts, developers, classes, apps, levels, tasks and all score records.

The second objective that must be achieved is the ability a user must have to add data to the database. This operation, just like the previous one, should be only possible if the type of user that made the request to add data is allowed to do so. Analyzing all entities present in the database that can access the web application the following requirements were made:

- Student – since the application revolves around scores, every student should have the ability to add score records to the database, a process that is done via the mobile games that are registered in the database. Students shouldn't have the ability to add anything else to the database;
- Teacher – the teacher does not have any need to add any data to the database and, as such,

shouldn't have the ability to do so;

- Expert - just like the teacher, the expert also does not need to any data to the database, an ability that is also inexistent to him;
- Developer – initially it was decided that the developer should be able to add any student, class, teacher, expert and developer to the database. However, after some thought and further testing, it was decided that the developer could only add other developers to the database. The other classes were assigned to the database administrator, someone that has direct access to the database itself.

The third objective is related to the ability to remove requested data, an ability that, just like the previous objectives, should only be possible to a user that is allowed to do it. As such, all entities that can access the web application were analyzed and the following requirements were determined:

- Student – the student should not have the ability to remove any data from the database;
- Teacher – the teacher should not have the ability to remove any data from the database;
- Expert – the expert should not have the ability to remove any data from the database;
- Developer – just like with the previous requirement, it was firstly decided that the developer should be able to delete any student, class, teacher, expert and developer from the database. Those functions were all assigned to the database administrator as well, and the developer only could retain the ability to remove developers from the database.

The last objective is aimed at the ability to alter data present in the database. Just like all the other objectives, this can only be done if the user that wants to do said operations is allowed to do so. So, the permissions allowed for each type of user that can access the web application should be as follows:

- Student – the student should only have the ability to change his password that serves has one of the access parameters to the web application;
- Teacher – the teacher should only have the ability to change his password that serves has one of the access parameters to the web application;
- Expert – just like with the teacher and the student, the expert should only have the ability to change his web application access password;
- Developer – the developer should have the ability to change the any detail of any student, teacher, expert and developer, including the password mentioned in the previous bullet points, except the identification codes of any of them.

With the listing process of all the requirements the application programming interface needs, the next step is their validation, a process that is needed to ensure that every requirement is needed and that they achieve what they are intended to achieve.

5.4. Requirements validation

Regarding the first objective, all requirements listed are in accordance to it and allow the access to any data that is needed to develop the web application features that were envisioned, without giving access to data that should not be accessible.

Regarding the second objective it, all requirements that were listed fulfil it, and allow the data collecting process, as well as add any user without compromising the integrity of the service that is provided.

The requirements that were listed whilst having the third objective in mind, although simple, fulfill it in a way that, just like requirements mentioned for the second objective, do not compromise the integrity if the service this project aims to provide.

For the last objective, all requirements are also in accordance to it, and allow users to alter the data they should be able to alter without compromising other existing data and the identity of any user present in the platform.

With all the requirements listed and validated the building process of the application programming interface could begin, a process that started by modeling the system and the features that would be implemented.

5.5. Modeling

The modeling of the application programming interface was a process that aimed to achieve an easier time when the development of the component itself came to be. Since the features that were planned to be expected were relatively simple and there was no graphic design the modeling itself was expected to produce a final result that was simple as well.

The model that was decided to be used consisted of four diagrams, one for each type of user that can access the web application so that they can be less cluttered. Use case diagrams are UML modeling language diagrams that represent a project or system from the user's perspective by showing use cases, sets of actions that a system can or needs to perform. These diagrams should be simple since they do not specify the details of the actions a system can perform nor the user's details. According to this definition and the list of requirements made and validated in the previous sections this diagram was the natural choice for modeling the application programming interface [103], [104].

Just like the models made for the database, the software program that was used for modeling was Visual Paradigm since it was already used and contained support for the use case diagrams that were planned to be designed.

The process for designing each of the diagrams consisted of analyzing the entities that could access the web application and the list of requirements made in order the determine the features the application programming interface would have implemented for each type of user.

As such, for the entity representative of students the following diagram (Figure 7) was developed:

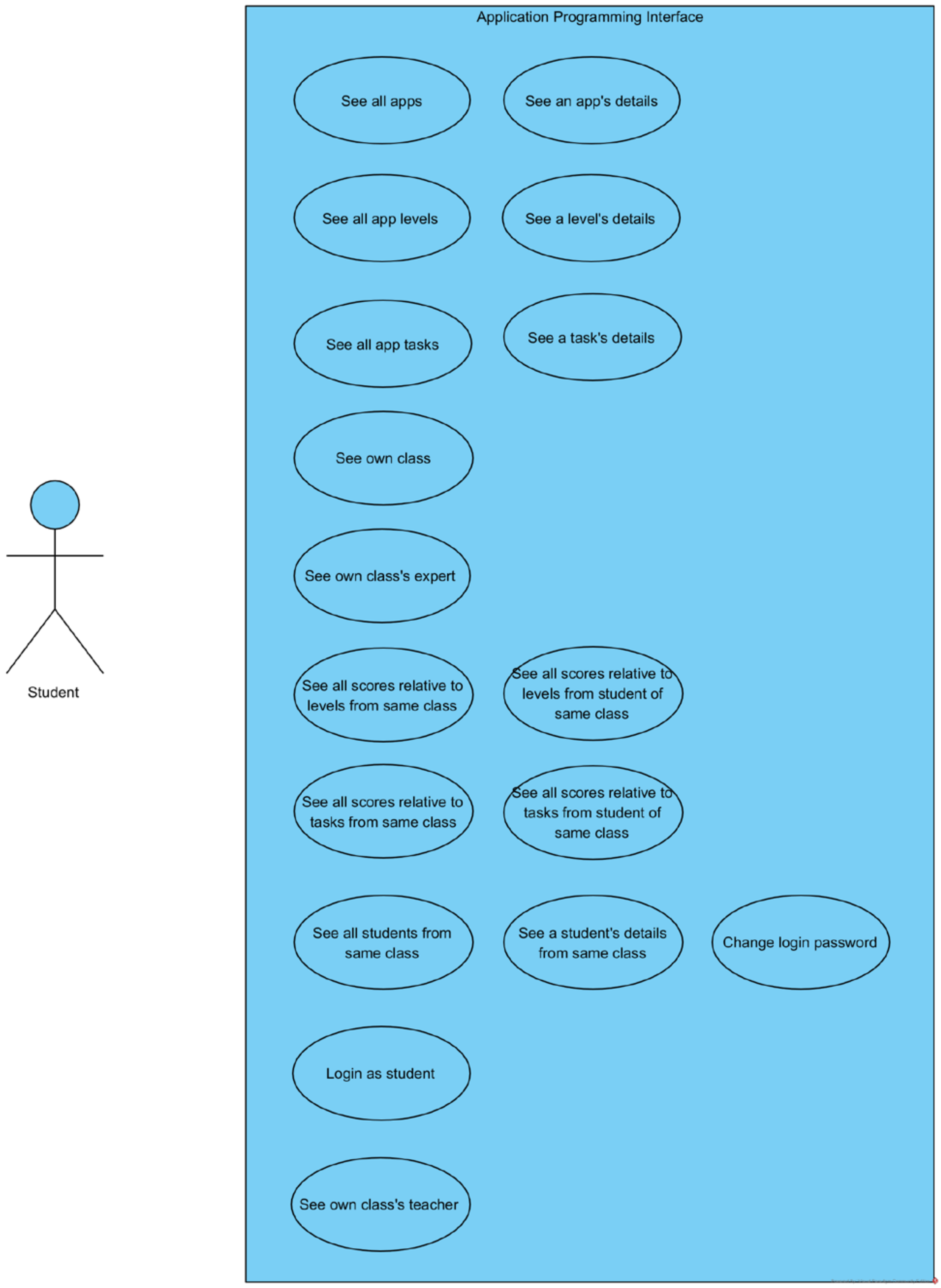
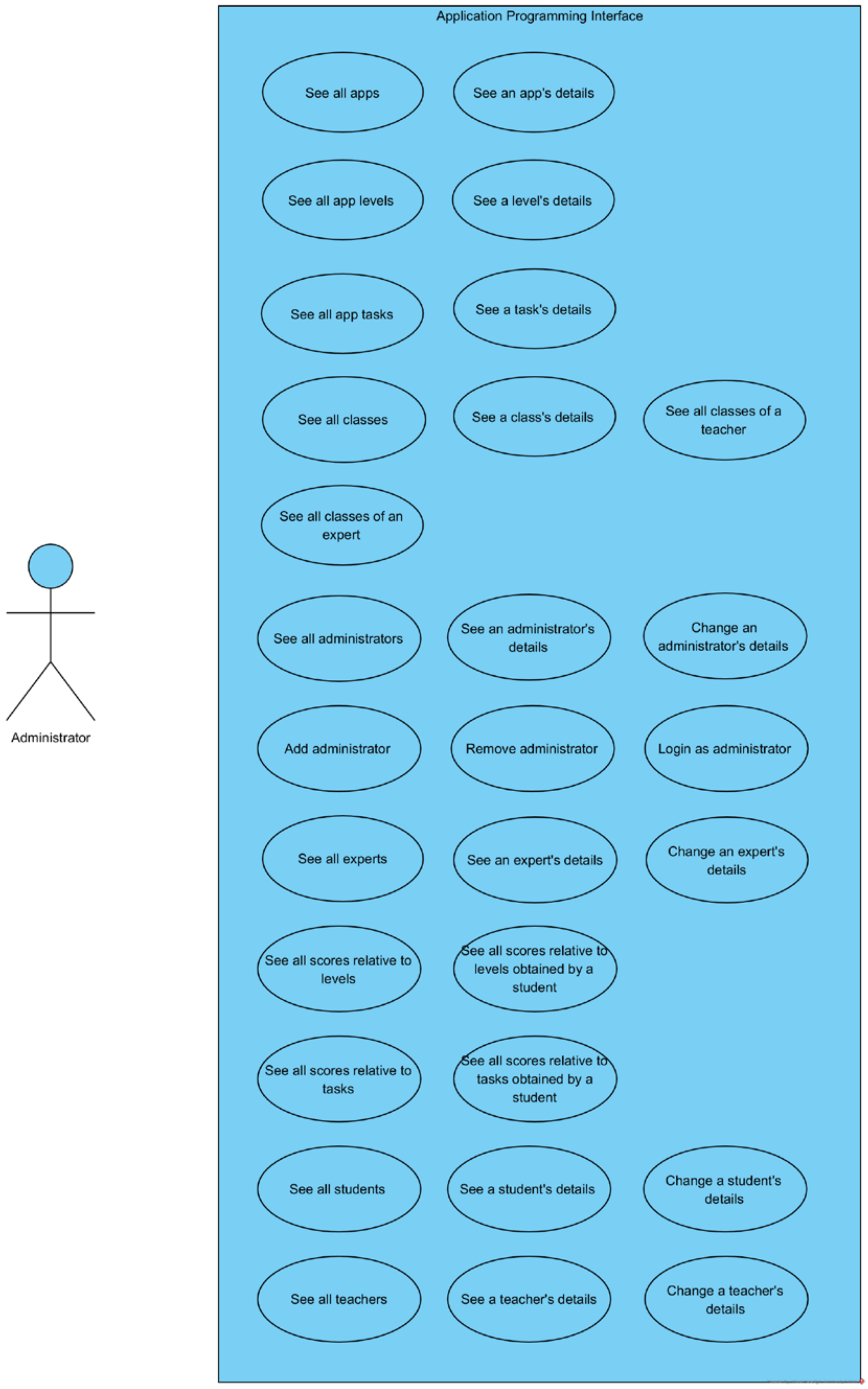

Application Programming Interface
See all apps
See an app's details
See all app levels
See a level's details
See all app tasks
See a task's details
See own class
See own class's expert
See all scores relative to levels from same class
See all scores relative to levels from student of same class
See all scores relative to tasks from same class
See all scores relative to tasks from student of same class
See all students from same class
See a student's details from same class
Change login password
Login as student
See own class's teacher
Student

Figure 7 - API Student Use Case Diagram (direct connections removed for visibility)

Before discussing the diagram there was a change that was made to the diagram type. Use case diagrams include the connection between the actor, in this case, the student, represented by a stick figure, and each use case or functionality. Due to simplicity's and readability's sakes and since all use cases were directly connected to the actor without any kind of more complex connections, these were removed.

Moving on to the diagram, it is possible to see that every requirement that was determined in the previous sections was taken into account when designing this model and, having them as a starting point, some other functionalities were thought of. These include the ability to see the details of the app, app level, app task, class, expert, level score, tasks score, student and teacher tables, having in mind the restrictions that were previously set. These restrictions only allow the student to see his own class, the students, expert and teacher of his own class and the scores belonging to students of his own class whilst all other tables can be seen in their entirety. Another function that was added was the ability to login into the web application, a function that was also implemented for the other user types, having in mind their attributes.

For the administrator user type the following use case diagram (Figure 8) was designed:



Application Programming Interface
See all apps
See an app's details
See all app levels
See a level's details
See all app tasks
See a task's details
See all classes
See a class's details
See all classes of a teacher
See all classes of an expert
See all administrators
See an administrator's details
Change an administrator's details
Add administrator
Remove administrator
Login as administrator
See all experts
See an expert's details
Change an expert's details
See all scores relative to levels
See all scores relative to levels obtained by a student
See all scores relative to tasks
See all scores relative to tasks obtained by a student
See all students
See a student's details
Change a student's details
See all teachers
See a teacher's details
Change a teacher's details
Administrator

Figure 8 - API Administrator Use Case Diagram

Just like the previous diagram, the requirements that were determined were taken into account when designing this diagram. The administrator will have the ability to see every table and its details, an ability that only is attributed to him. Two new functionalities were also included, the ability to see the classes of a teacher and the ability to see the classes of an expert, functionalities that were thought of as useful for the development process of the web application. The administrator can also add, remove and alter administrator table entries, abilities that only exist together on this table and only be done by the administrator.

The diagram that was put together for the teacher user type (Figure 9) was the following:

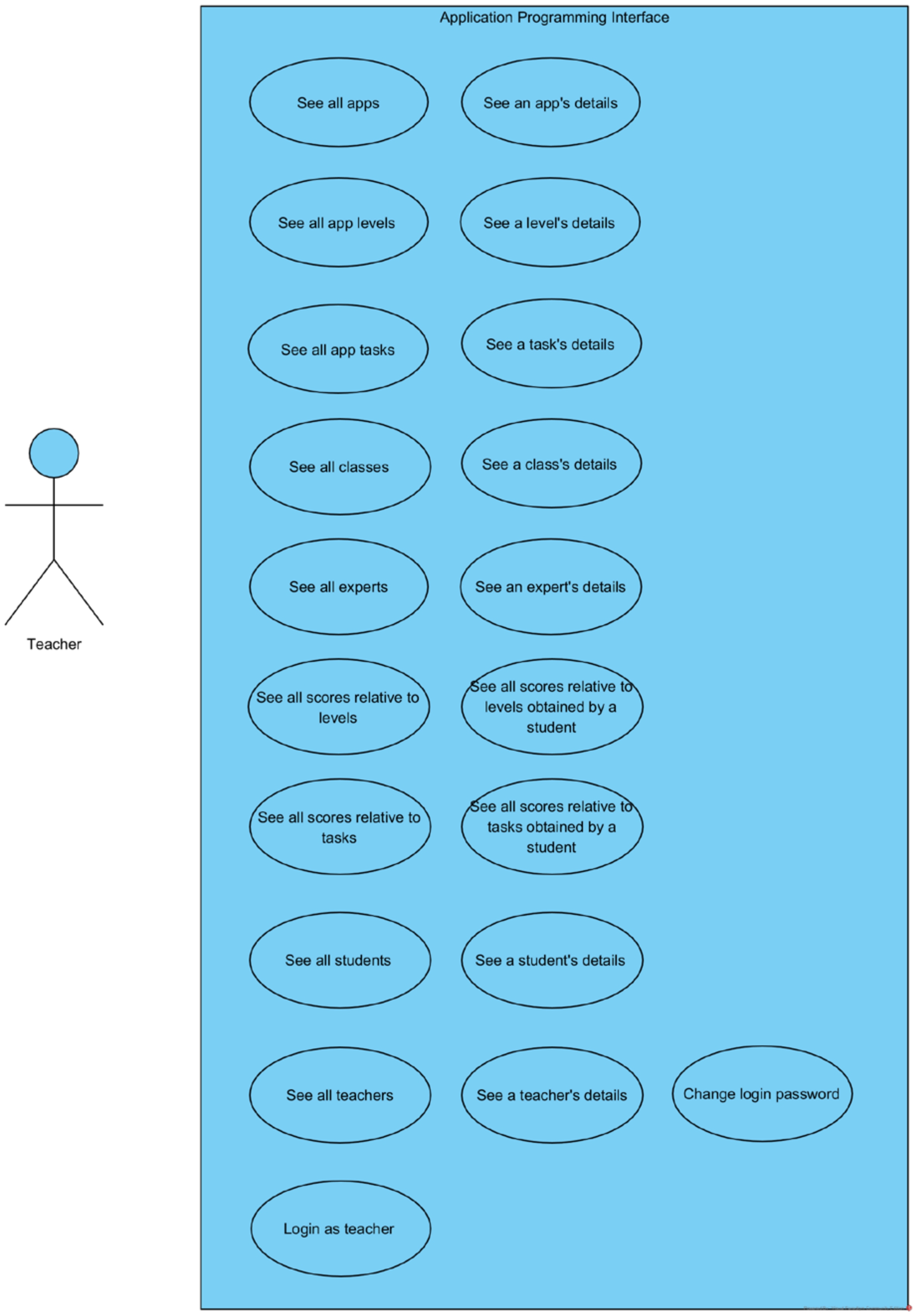


Figure 9 - API Teacher Use Case Diagram (direct connections removed for visibility)

As can be seen in this diagram, the teacher has a few more functionalities available to him and less restrictions than the student. For example, while the student can only see his own class, its own teacher and expert and the scores from his classmates, the teacher can see all those table and their details. All other functionalities were made taking into account the requirements that were listed.

The last diagram (Figure 10), developed for the expert user type, was the following:

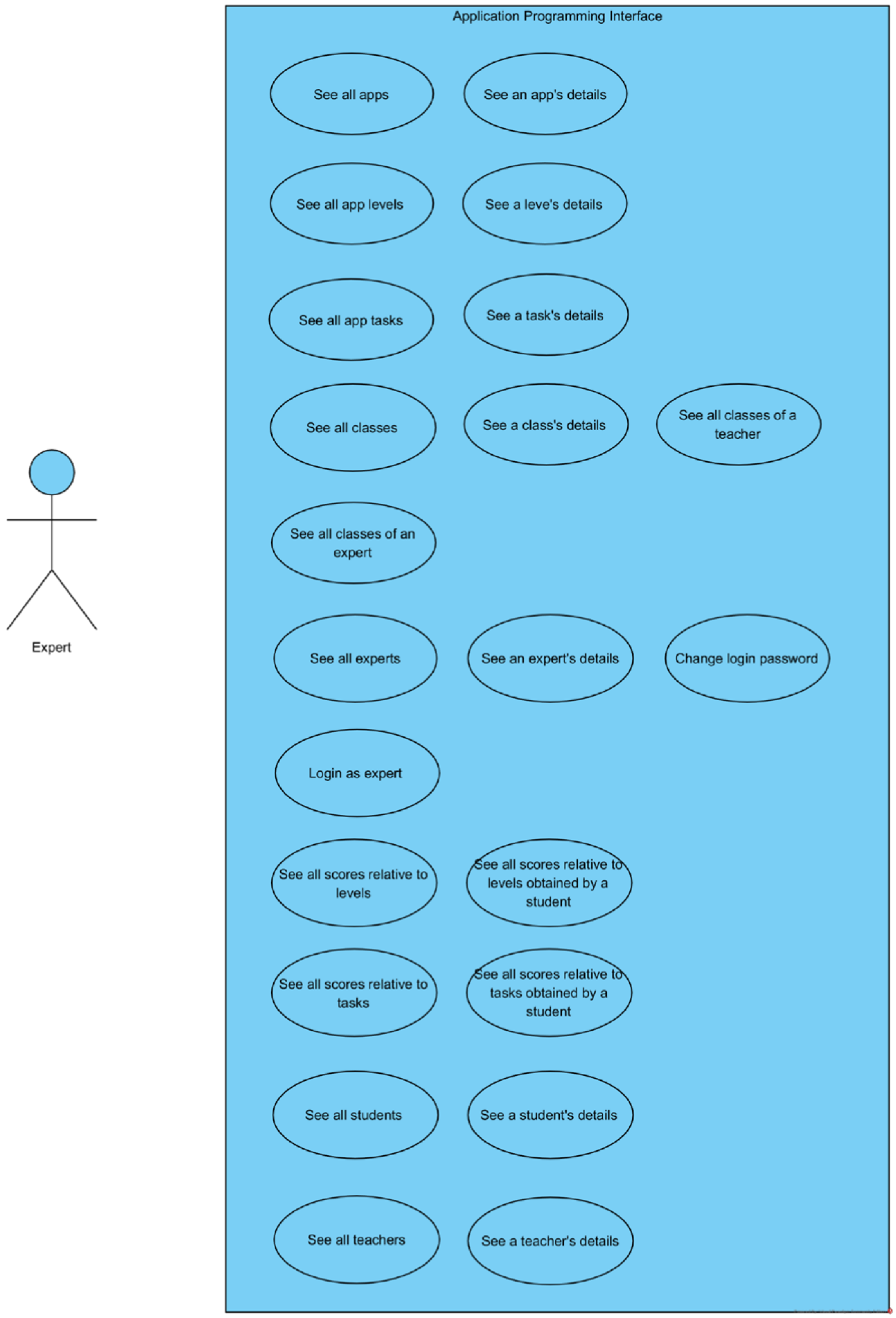


Figure 10 - API Expert Use Case Diagram (direct connections removed for visibility)

The expert diagram has a few less retractions than the teacher table, although it is quite similar to it. The main difference is the ability to see the classes of a teacher and the classes of an expert. These functionalities were thought of as useful for this user type since there were some functions that were planned for the web application, informally at this point, that could use these abilities, functions that are explained in a further section. Regarding the teacher diagram, it is possible to see that it does not include these functions thanks to the fact that no functionality of the web application that could use it was thought of.

With the model of the application programming interface system and the features that were planned to be developed the next step was the building process of the component itself.

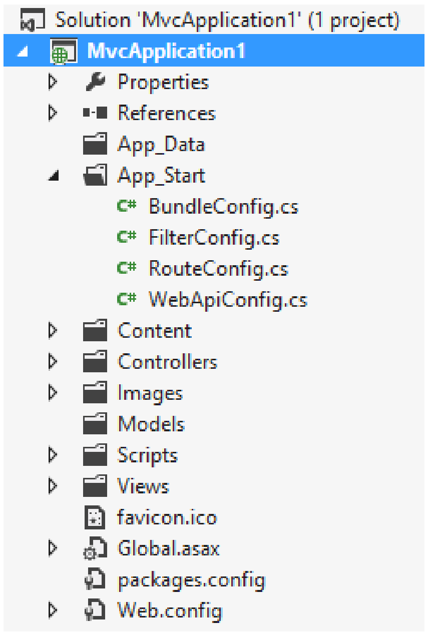
5.6. Implementation

The application programming interface process was not a very complex and although the features that would be implemented were not very complex or extensive, the process of developing the application programming interface itself was extensive since the features to be implemented were plentiful.

Each one the features were in great part consisting of the building of a SQL Query and a request made to the database regarding its data that includes the user access parameters and, in some cases, like when the user requests a single record of table, for example, a teacher, parameters that identify the data the user wants.

Like it was mentioned in the Technologies and methods section the chosen technology for this component was ASP.NET and the chosen programming interface was Microsoft Visual Studio. Since the computer that was provided by professor Victor Alves was running on an old operating system, Windows Server 2003, the hosting software, Internet Information Services, also mentioned in the same section as the previous software technologies, was only available in an old version. As such, and since the most recent version of ASP.NET at the time of the development of the application programming interface was found to not be usable with it, an older version had to also be used. The most recent version of ASP.NET that was tested and found to be compatible with the hosting software was ASP.NET 4.0, a version of the technology that was only possible to develop in using a version of Microsoft Visual Studio that was also not the most recent one at the time of the development of this project, Microsoft Visual Studio 2012.

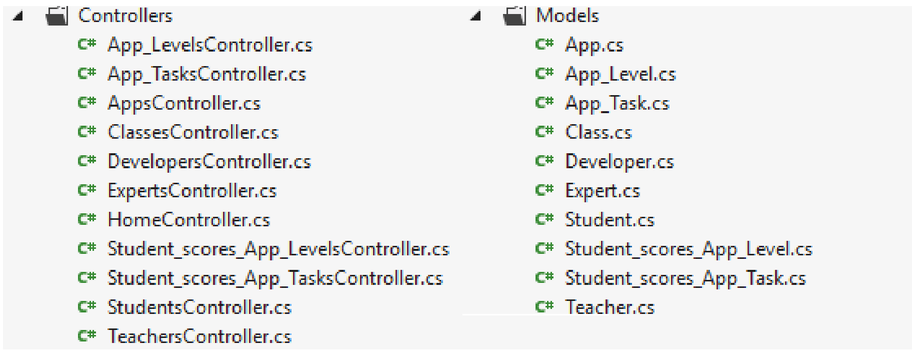
The initial structure of the project (Figure 11) was built by using the ASP.NET 4 MVC Web Application project preset with the Web API template which created a file directory with the following structure:



```
Solution 'MvcApplication1' (1 project)
  MvcApplication1
    Properties
    References
    App_Data
    App_Start
      BundleConfig.cs
      FilterConfig.cs
      RouteConfig.cs
      WebApiConfig.cs
    Content
    Controllers
    Images
    Models
    Scripts
    Views
    favicon.ico
    Global.asax
    packages.config
    Web.config
```

Figure 11 - Initial API structure example

The main focus points of this project were the folders called Models and Controllers (Figure 12) and the file called WebApiConfig.cs present of the App_Start folder. The Models folder contains all the tables present in the database as well as their structure in the C# programming language, using classes. The Controllers folder contain a file for each of the classes present in the database just like the Models folder. Each file, which makes use of the classes present in the Models folder, is responsible for implementing the features the application programming interface has available. The WebApiConfig.cs file, which use is not necessary in the most recent versions of ASP.NET, has the purpose of assigning functions to routes and request types, both of which are discussed further in this section.



```
Controllers                                  Models
  App_LevelsController.cs                      App.cs
  App_TasksController.cs                       App_Level.cs
  AppsController.cs                            App_Task.cs
  ClassesController.cs                         Class.cs
  DevelopersController.cs                      Developer.cs
  ExpertsController.cs                         Expert.cs
  HomeController.cs                            Student.cs
  Student_scores_App_LevelsController.cs       Student_scores_App_Level.cs
  Student_scores_App_TasksController.cs        Student_scores_App_Task.cs
  StudentsController.cs                        Teacher.cs
  TeachersController.cs
```

Figure 12 - API folder structure

Having developed all the features that were planned for the application programming interface component they had to be assigned to a route and a request type, a process that was done by editing the preexisting file WebApiConfig.cs. A route is part of the address that is used to access the application programming interface that is bound to one of its features, alongside the type of request made, for example, if the public address of the application programming interface is http://www.test.com and the route for seeing all students is /api/students, all students can be seen by accessing or requesting said data from the address http://www.test.com/api/students.

The application programming interface works by responding to requests it receives. The type of requests that it receives are HTTP requests which are made according to the HTTP protocol responsible for structuring request and responses between two entities. The types of HTTP requests implemented for the application programming interface are GET, POST, DELETE and PATCH. These requests are made for requesting data, requesting to add data, requesting the deletion of data and requesting changes to data present in the database, respectively [105].

During the development process of the application programming all HTTP requests were made into POST requests due to some incompatibilities that occurred when trying to send a request with all the information the web application needs to provide the requested data. Since there were no functions assigned to the same route but with HTTP request type there was no problem making this change.

Having the application programming interface built and ready to be available it needed to be hosted so it could be accessed. This process is discussed in the next section.

5.7. Hosting

Since, at this point of the development process, the application programming interface is ready to be deployed and made available, it must be prepared for such. So, in order for it to be possible to use, it had to be built in a format that could be usable for the hosting software, Internet Information Services.

By using Microsoft Visual Studio that process was easily achievable and fast, resulting in a folder with files that could be hostable. That folder was hosted and, by granting the necessary permissions for it to be sharable with anyone who tries to access it, it was available through a public address on the internet.

5.8. Testing

Just like with the database, having the application programming interface deployed and available publicly, the necessary step to ensure that every aspect of it was working as intended and without any problems was to test both the application programming interface and the hosting aspect of it.

The initial testing, a process that was done during the development process of several features until the test of all the features after the application programming interface was deployed, consisted of testing each created route and request, multiple times. All requests related to data visualization could be tested by accessing the route through a browser but to test the remaining types of request that method does not function. As such, a software program called Postman directed towards testing application programming interfaces by allowing users to create and send different types of request with any data of their choosing and analyzing server responses was used for the great majority of the testing process of this component of the project. All data that was involved in this process was random, created only for testing purposes, and was not based on any kind of data that previously existed [106].
All the remaining tests that were done were in its majority done during the development and testing phases of the web application itself and the mobile game that was mentioned in the Database testing section, a game that makes use of the database through the application programming interface of this project to store the data it collects from students.
Similarly to the database, there was no major change done to this component after the testing phases, a fact that was mostly due to the planning phases that were discussed in first sections related to this component and the straightforwardness of the features that were implemented.
At this point of the development process of this project, all the conditions required to fully develop the web application were gathered since the database was ready and a way to communicate with it was established successfully, a process that is discussed in the following sections.

6. Web application

Since both the database and application programming interface are ready for use the remaining component is the most important one, the web application itself. It was decided that it was going to be the last component to be built since it need both of remaining components.

The technology that was decided for building the web application was Angular and, just like the application programming interface, the one that was selected to host it in order to make it available on the internet was Microsoft's Internet Information Services. This section explains these two processes and discuss the building and planning phases of the web application.

Just like the development process of the application programming interface the planning of the general steps of the construction of the web application was the first objective of this process in order to have most efficient building process as possible and just like the application programming interface the starting point for this were the steps for developing the database. As such, the steps for developing the web application were planned as follows:

- Case study analysis – the analysis of the purpose and goals the web application must achieve;
- Requirements analysis – the listing of every requirement the web application must fulfill;
- Requirements validation – the verification process of the list of requirements made in the previous step;
- Web application sketching – the process of building models that will serve as the main inspiration for creating the web application;
- Web application implementation – the implementation process of the web application in accordance with the list of requirements made and the models created in the previous step;
- Web application hosting – the hosting process of the web application in order to make it available through a public internet address;
- Web application testing – the testing and correction process of potential problems the web application or its hosting component might have [84]–[86].

6.1. Case study

Since at this stage of the development process the database and application programming interface were deployed and available through a public address on the internet, every required component was ready to start building the web application itself, the primary objective of the project.

The main objective of the web application is to have a way to show information regarding data collected from mobile games developed towards primary school students (junior school in the United Kingdom or elementary school in the United States of America). That information is showed in a way that makes it easier to identify and understand potential cases of dyscalculia, as well as the areas of mathematics and arithmetic where it is more prevalent.

As such, it should have a visually appealing and easy to understand interface so it its data can be analyzed and conclusions made about them by any user with access to it, be it a student, a parent, a teacher or an expert.

That being, the development process must be meticulous and thought out since it is the culmination of all the work up to this point of the project and the work that was yet to come. That process is discussed in the following sections.

6.2. Goals

As mentioned, the primary objective of the web application is to show data collect by dyscalculia related mobile games directed towards young students. That being, the goals for this last component of the project should be defined while having in mind the ease users should have to view and understand that data.

As such, the objectives that were defined for web application were:

- Allow students see all scores that were achieved and analysis of the performances that were had in each one, and in general;
- Allow teachers to see his classes, as well as his students, and analyze each one in their individuality and in general;
- Allow experts to see all students and classes, and analyze each one in their individuality and in general;
- Allow administrator to see and edit any user type as well as add and remove other administrators;
- Every analysis should be easily understandable by any user that has access to the web application.

After the main objectives for these components were listed the requirement list was enumerated, a process that is explained in the next section.

6.3. Requirements analysis

For this step of the project, all objectives were analyzed and the requirements that were necessary to be fulfilled for the web application to work as intended were defined.

As such, the first objective consisted of making the web application capable of showing students all scores that were achieved in any game, as well as an analysis of each one and an analysis of the student's performance.

As such, for this objective, the requirements that were determined are as follows:

- Allow the students to see a general analysis of the performance had in different areas of dyscalculia;
- Allow the students to see a more thorough analysis related to the general performance that was had, including the results of other students of his class;
- Allow the students to see all the results that were achieved, including scores and times in each level and task of any game;
- Allow the students to see an analysis of each score.

The second objective is focused towards the teacher and making it possible to see all of his classes and students. For each student and class, the teacher should also be able to see an analysis in order to understand their performance.

So, for the second objective, the requirements that were made are as follows:

- Allow the teachers to see all of his students;
- Allow the teachers to see all of his classes, as well as the student's that belong to each class;
- Allow the teachers to have access to the same analysis a student has of his performances;
- Allow the teachers to have access to an analysis of all of his classes' performances;
- Allow the teachers to see an analysis of each individual class's performance.

The third objective is related to expert's side of the web application. The expert should be able to see all students and classes and analyze them, either in general or in their individuality.

As such, the requirements made for this objective are the following:

- Allow the experts to see all students that exist in the platform;
- Allow the experts to see all classes available, as well as the student's that belong to each one;
- Allow the experts to have access to the same analysis a student has of his performances;
- Allow the experts to have access to an analysis of all of his classes' performances;
- Allow the experts to see an analysis of each individual class's performance.

The fourth objective is related to the understandability of all the analysis that can be presented. As such, and since they are all based on the same kinds of results, an idea was had to make them, as much as possible, in graph form, with only a written explanation of the results they possess. So, a graph had to be chosen at this point, and the type of graph that was chosen was the radar chart. This kind of graph is helpful to display data belonging to different variables. For example, it is helpful, in the case of this project, to show the percentage of questions the student got right in an area of dyscalculia. If a student had 75% of the tasks of the algebra area, 90% of the calculus tasks and 50% of the measures tasks they are easily shown and easy to understand [107].

Having made the listing of requirements the next step was to assure that they were in compliance with objectives that were defined for the web application.

6.4. Requirements validation

Regarding the first three objectives, all the requirements that were listed for each of the user types are all serving the main objective in a simple and straightforward way and allow access to any information present in the platform in a simple and understandable way. They also served as a mean to start forming an idea of what the web application became.

The requirement that makes up the last objective focuses on the way that all analysis are done, namely the way the data is presented, and not only makes said objective possible, but also serves as a very good aid in achieving all the analysis related requirements that constitute the other objectives.

Having these requirements ready, due to them being determined and validated, the next step is to start modeling the web application, namely it's graphical interface, the screens that the users that connected to the web application is presented.

6.5. Modeling

Since the web application was the most complex component of this project, the modeling process of it was also more extensive than those of the other components due to the fact that it should contain more information about the way its construction was going to occur. As such, the modeling process of the web application was initially divided into two parts, the design of the functions that were going to be available and the design of the actual user interface of the component, in that order.

The first part was similar to the modeling process of the application programming interface, a process that focused on showing the functions that were available for each type of user with access to the application.

Just like the application programming interface, the model that was going to be developed consisted of four use case diagrams, one for each type of user that can access the web application. As a reminder, use case diagrams are UML modeling language diagrams that represent a system from the user's perspective by showing use cases, a name used to represent actions that a system can or needs to perform. As such, it was the natural selection of diagram for modeling the functions the web application would have implemented. And, just like most diagrams made during the planning phases of the project, the software that was used was Visual Paradigm, a program that was also already used to develop use case diagrams [103], [104].

The modeling of the function designing part for the web application was done firstly by analyzing all users that could access the web application and the list of requirements that was defined in the previous sections. This was so that all the features that were to be implemented for each type of user were clear and easy to understand in the final result in order for every detail to be accounted for when it came the time for them to be developed.

As such, for the student user type, the following diagram (Figure 13) was developed:

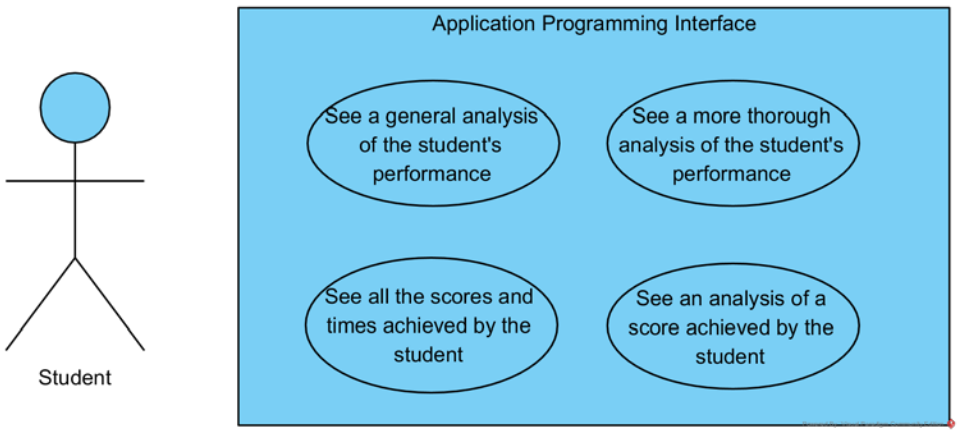

Application Programming Interface
See a general analysis of the student's performance
See a more thorough analysis of the student's performance
See all the scores and times achieved by the student
See an analysis of a score achieved by the student
Student

Figure 13 - Web Application Student Use Case Diagram (direct connections removed for visibility)

Again, just like the use cases made for the application programming interface, the use case diagram that is shown as well as the following ones, suffered a change regarding its usual format. Since all connections between the actor, in this case, the student, and each use case or functionality were simple and direct and no complex connections exist, they were visually removed for readability's sake.

As is possible to understand, a student that logs in into the web application should have access to functions that are described. These functions are the focus of the web application that should make possible for the student to understand his condition regarding dyscalculia. As such, when designing these functions, the first visual aspects of the web application were being thought of at the same time.

When logging in to the platform, the student should be able to see a general analysis of his performance on different areas of dyscalculia. If some further analysis was required, there should also be a way to access it in order to understand other details of the student's performance. Besides that, there should also be a way for the student to see all of the scores and times achieved on different tasks and analyze said parameters.

For the administrator user type the diagram that follows (Figure 14) was designed:

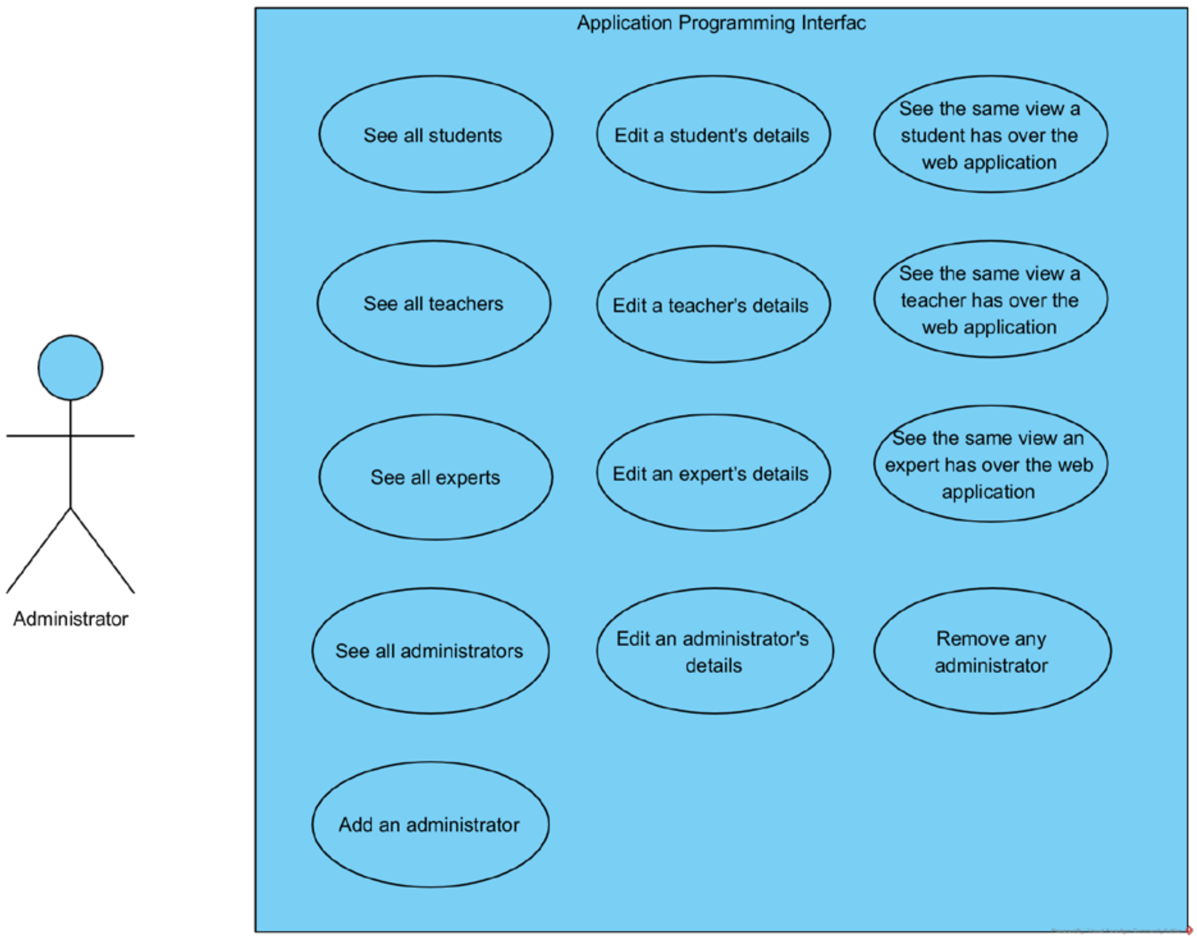

Application Programming Interfac
See all students
Edit a student's details
See the same view a student has over the web application
See all teachers
Edit a teacher's details
See the same view a teacher has over the web application
See all experts
Edit an expert's details
See the same view an expert has over the web application
See all administrators
Edit an administrator's details
Remove any administrator
Add an administrator
Administrator

Figure 14 - Web Application Administrator Use Case Diagram (direct connections removed for visibility)

The first idea when designing this diagram was to divide the administrators functions into four different sections: the students, the teachers, the experts and the administrators. Each of these sections should be accessible by clicking some element present in the initial page.

By analyzing each function, the first conclusion that was reached was that each starting page for each section should contain the all users that if focuses on. Each of these users should be clickable, except for the administrators, as can be seen, or allow, in some other way, to access that user's view over the application. Next to the user should be an image, or another element, in order to make a pop-up appear or redirect the administrator to a page where he can edit the details of the selected user. In the case of the administrators there should also be a similar way to remove or add any administrator.

The third diagram (Figure 15), put together for the teacher user type, was the following:

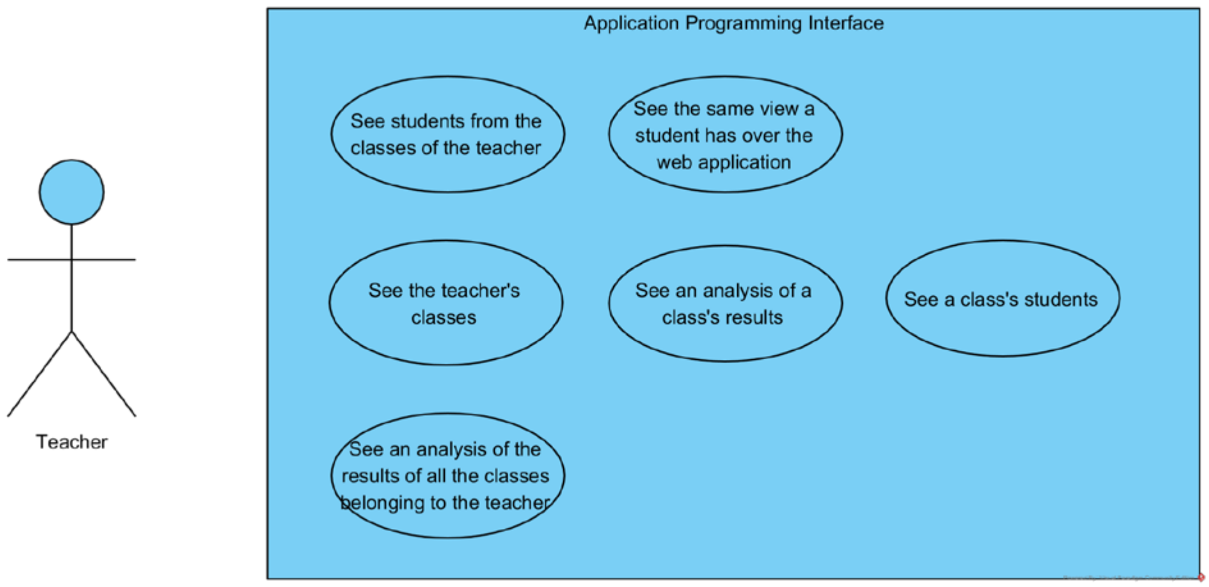

Application Programming Interface
See students from the classes of the teacher
See the same view a student has over the web application
See the teacher's classes
See an analysis of a class's results
See a class's students
See an analysis of the results of all the classes belonging to the teacher
Teacher

Figure 15 - Web Application Teacher Use Case Diagram (direct connections removed for visibility)

Again, for the teacher's functions, there was a decision made to divide them into different sections. These sections are three and as follows:

- Students, where the teacher should be able to see all of the students belonging to his classes and, just like with the administrator's case, click on them, or in some other way, to access the selected student's view over the application in order to see every result and analysis that was made;
- Classes, where the teacher should be able to see all of the classes he takes responsibility for and analyze their performance in different areas of dyscalculia, again by clicking on it, or in some other way, and also see the students that constitute it, where their view over the web application should also be possible to access;
- General results, where it is possible to see an analysis of the performance of all of the teacher's classes in different areas of dyscalculia.

The final diagram (Figure 16), the one that was design with the expert user type in mind, was as follows:

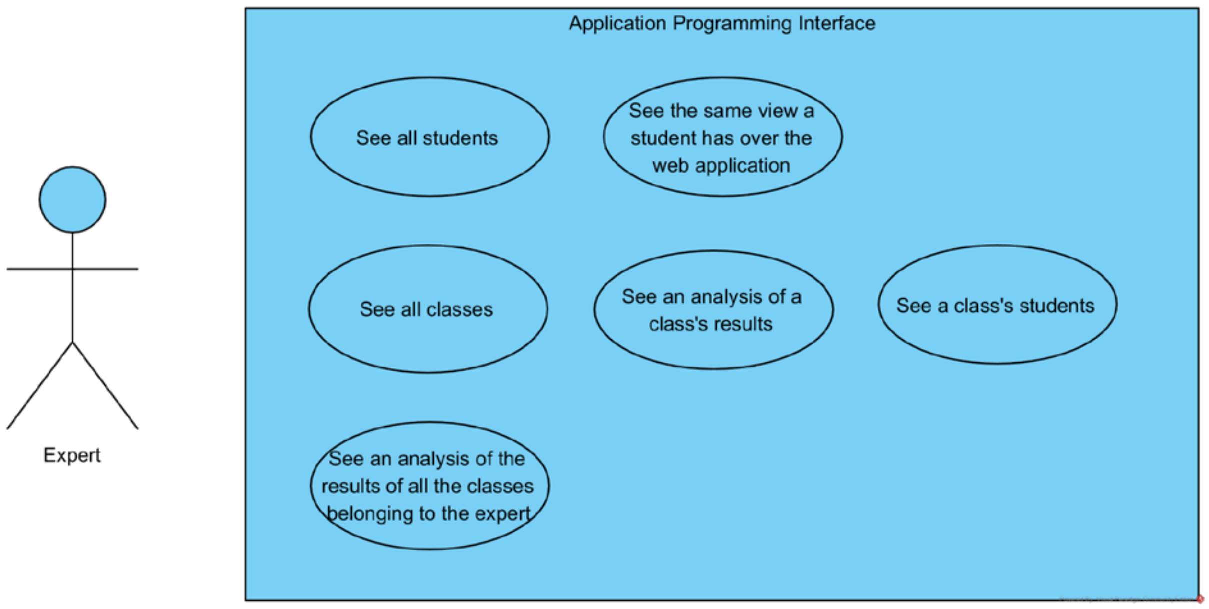

Application Programming Interface
See all students
See the same view a student has over the web application
See all classes
See an analysis of a class's results
See a class's students
See an analysis of the results of all the classes belonging to the expert
Expert

Figure 16 - Web Application Expert Use Case Diagram (direct connections removed for visibility)

The expert's functions are the same as the teacher's, the only different aspect being the number of students and classes each one can have access to. While the teacher only has access to his students and his classes, the expert has access to every student and every class.

In the first two sections, where the expert has access to all students and classes, the screen that is presented should contain a division between the expert's students and classes and all the others. This division should be done in order to allow a better access to the student or class that the expert wants to analyze, even though there was also an idea to implement a search mechanism to these sections.

The third section is exactly like the teacher's and should serve as a mean to analyze and understand the expert's classes' performance on different areas of dyscalculia.

Having developed these diagrams and formed an initial idea of what the web application should be, the first part of the modeling process was completed, where all functions available for all user types were defined and planned for the actual implementation of the web application itself. After that, the modeling of the actual graphical interface was in order, a process that was done using a modeling software program called Pencil, mentioned in the Technologies and methods section of this document.

Starting out with the first screen (Figure 17) a user is presented when accessing the web application:

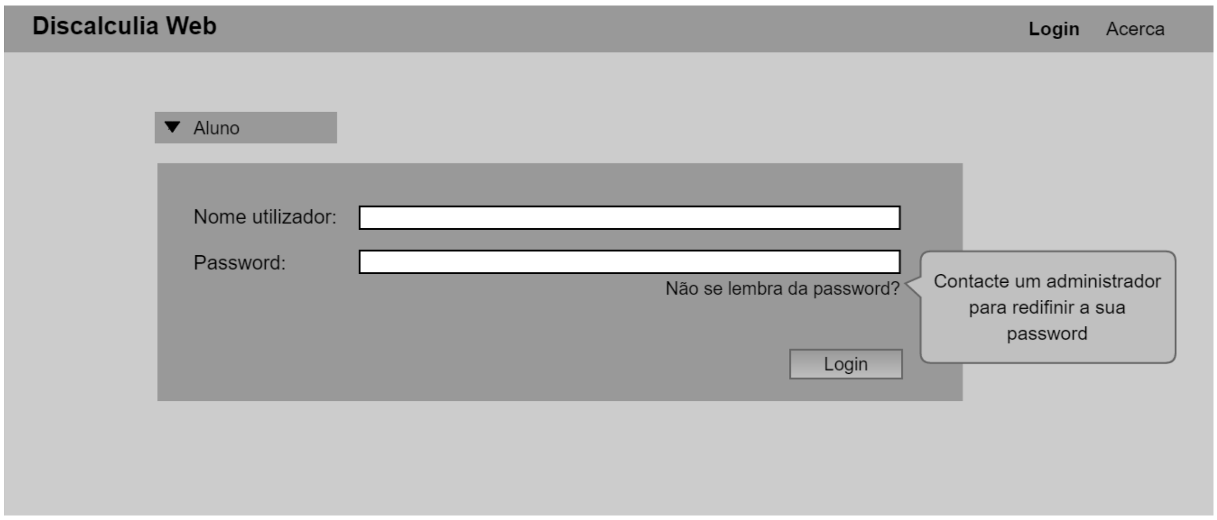

Discalculia Web
Login
Acerca
Aluno
Nome utilizador:
Password:
Não se lembra da password?
Contacte um administrador para redifinir a sua password
Login

Figure 17 - Initial screen model

In this screen, the user is presented two boxes, a box containing all the user type options that are possible to login into the platform as, and another box containing two fields where the user inserts the login parameters needed to access the web application, the username and the password field.

In the case of the user not remembering the login parameters needed to access the platform, there is a text that reads "Can't remember the password?" which, when hovered over it with the mouse pointer, makes a pop-up balloon appear on the screen that tells the user to contact an administrator of the web application in order for the password to be reset.

On the top left of the screen will be the name of the web application and, on the right side, there are two options that give the user access to the login page, the page presented in the previous figure (Figure 17) and to a page that explains what this application consists of, the "About" page.

The “About” page model is shown in the next figure (Figure 18):

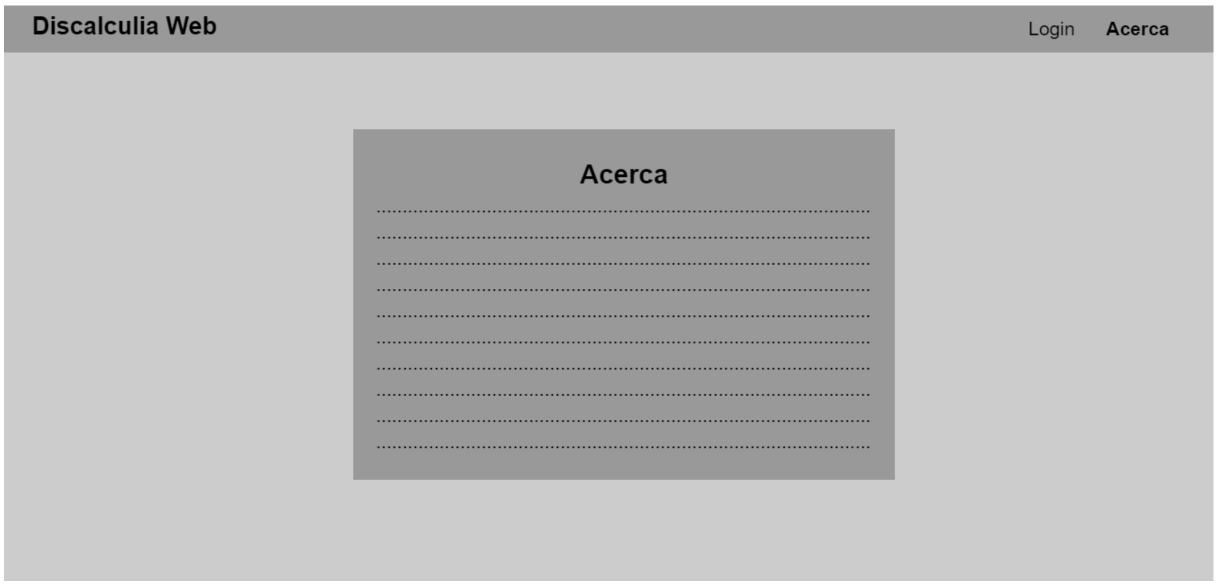


Figure 18 - About screen model

After clicking on the option that leads to this page, the user is shown a rather simple page that explains, as simply as possible, since the user that will access it probably does not fully understand all the concepts this project focuses on, what the web application is all about and what dyscalculia is. This text is not shown but it will be explained when discussing the final product.

The next model (Figure 19) focuses on the screen the user is presented if a student login is made:

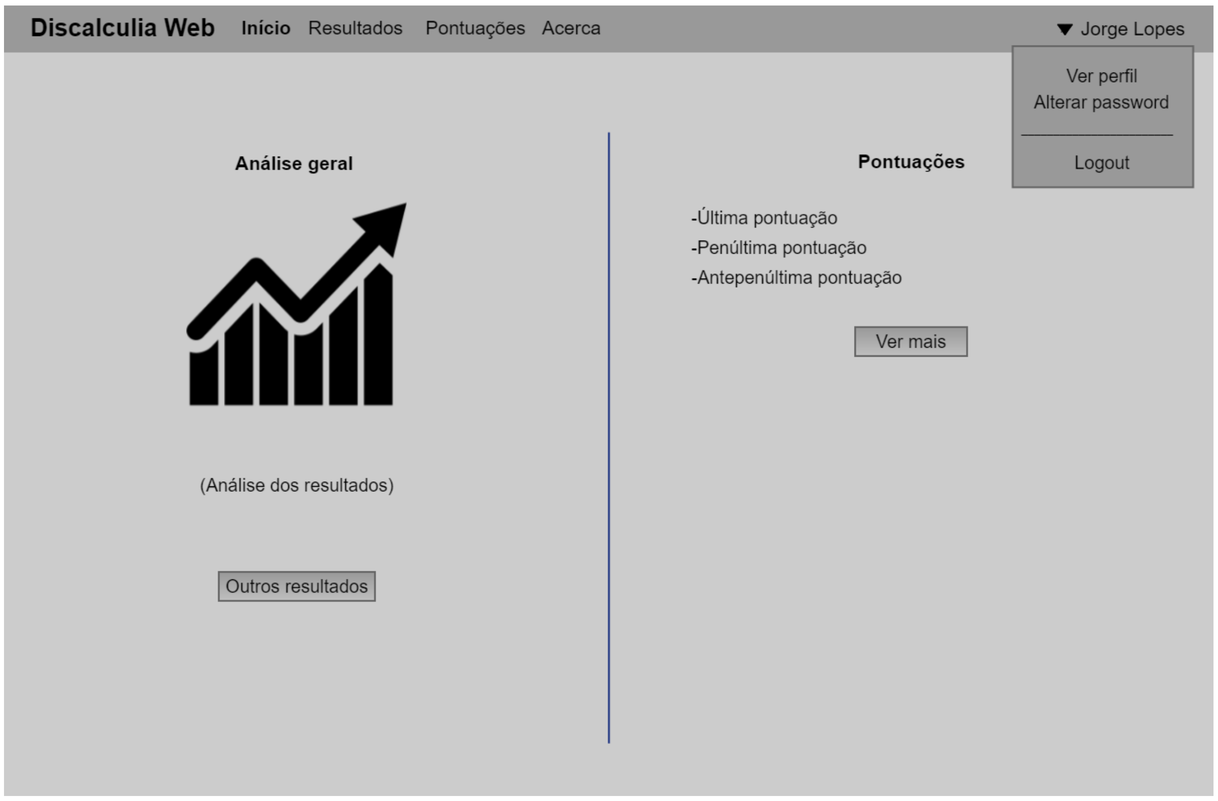


Figure 19 - Student initial screen model

When a user reaches this screen, he is presented some options. On the top left side of the screen the user as access to all the options regarding the pages he can access, pages that will be discussed in the following pages. On the top right of the screen the user is presented its name, a feature that is the same for all user type that login into the platform, which is clickable with the mouse cursor, and makes a pop-up appear with three options:

- An option to see the student's profile, a feature that will make another pop-up appear containing all the data that is stored in the database regarding the student itself;
- An option that also makes another pop-up appear and allows the user to change its password;
- A logout option which redirects the user to the first screen (Figure 17).

On the left side of the screen, the student can see a graph that shows results regarding all the scores that were obtained and, below it, a text that will be generated according to said results, that will explain the student if there are any detected dyscalculia symptoms. Below all that there is also a button that allows the student to see other results.

On the right side of the screen the student can see his last three scores that he obtained on any mobile game level, which are clickable and lead to a page that will allow the analysis of the student's performance. Bellow that there is button that allows access to a page that contains all the student's scores, a page that is explained further down.

When the student clicks on the button present on the left side of the screen or the *Resultados* option on the top, he is shown the following screen (Figure 20):

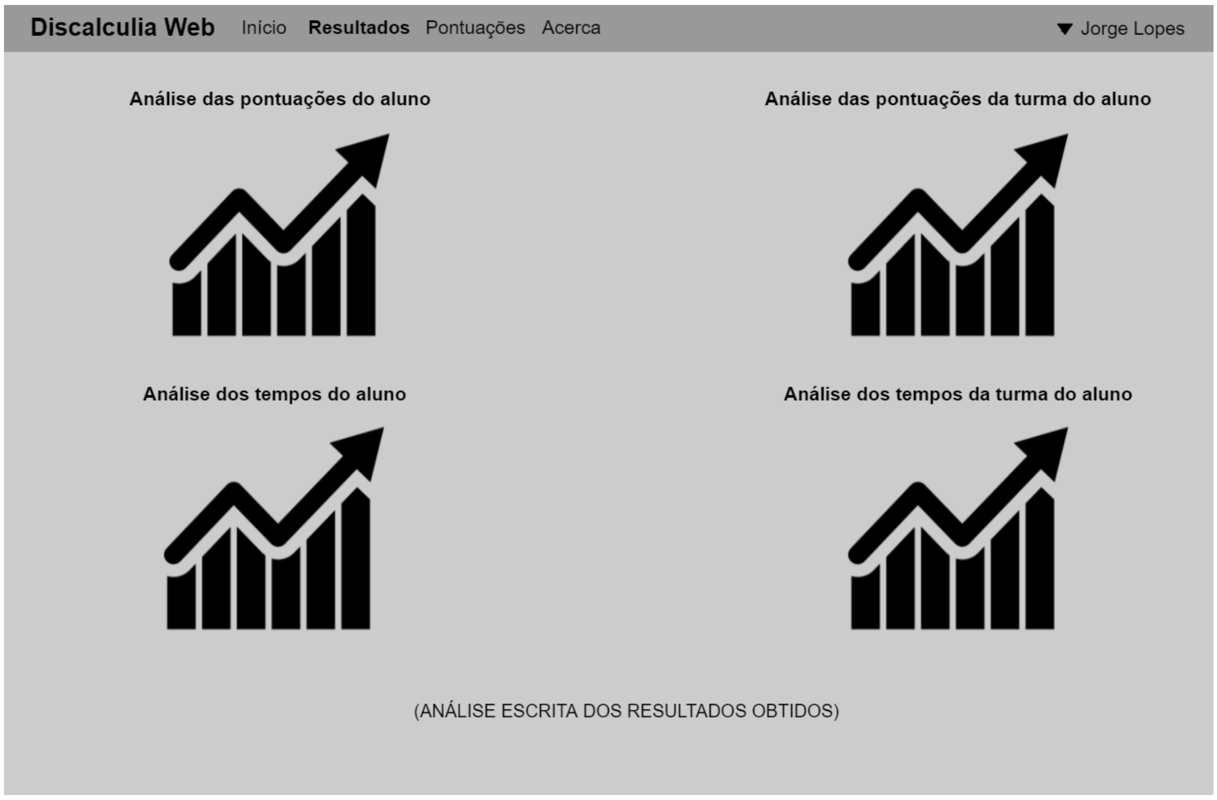
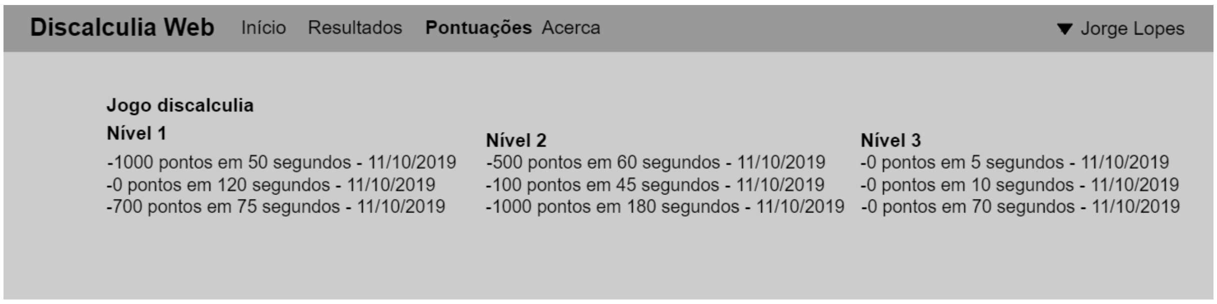

Discalculia Web
Início
Resultados
Pontuações
Acerca
▼ Jorge Lopes
Análise das pontuações do aluno
Análise das pontuações da turma do aluno
Análise dos tempos do aluno
Análise dos tempos da turma do aluno
(ANÁLISE ESCRITA DOS RESULTADOS OBTIDOS)

Figure 20 - Student general analysis screen model

This screen presents the user four graphs similar to the one present in the initial student screen (Figure 19). Two of these graphs are about the student's performance on different areas of dyscalculia, directed towards the scores and times that were achieved. The other two are the same but directed towards the student's class, which allow students to compare their performances with their classmates.

At the bottom of the screen there will be a text that will be generated according to the student's performance on each area of dyscalculia when compared with his class.

Coming back to the first screen (Figure 19), if the user clicks on the button on the right side of the screen or the *Pontuações* option on the top, the web application will present the following screen (Figure 21):


Discalculia Web Início Resultados **Pontuações** Acerca ▼ Jorge Lopes

Jogo discalculia

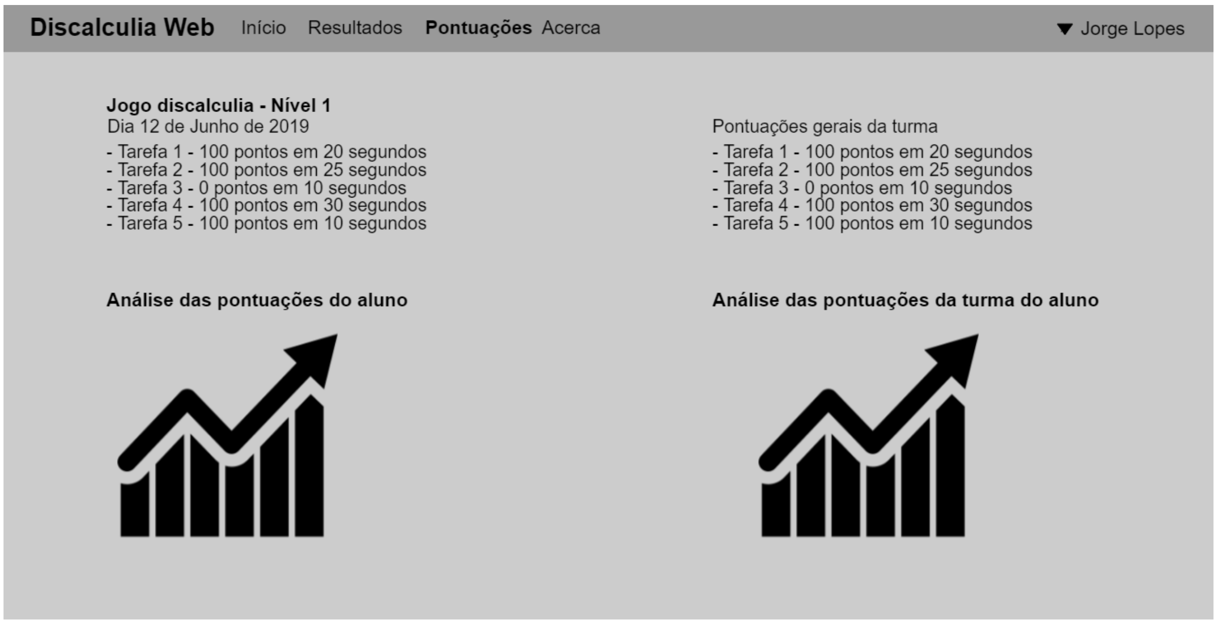
Nível 1
-1000 pontos em 50 segundos - 11/10/2019
-0 pontos em 120 segundos - 11/10/2019
-700 pontos em 75 segundos - 11/10/2019

Nível 2
-500 pontos em 60 segundos - 11/10/2019
-100 pontos em 45 segundos - 11/10/2019
-1000 pontos em 180 segundos - 11/10/2019

Nível 3
-0 pontos em 5 segundos - 11/10/2019
-0 pontos em 10 segundos - 11/10/2019
-0 pontos em 70 segundos - 11/10/2019

Figure 21 - Student scores screen model

In this screen the student will see his scores on all games registered on the platform and on each individual level. These scores will be clickable and will lead the student to page where he can see more details about his performance, so he can further understand it.
The following page (Figure 22) is accessible by clicking on any score in the previous page (Figure 21) of by clicking on any score present on the right side of the initial student screen (Figure 19):


Discalculia Web Início Resultados **Pontuações** Acerca ▼ Jorge Lopes

Jogo discalculia - Nível 1
Dia 12 de Junho de 2019
- Tarefa 1 - 100 pontos em 20 segundos
- Tarefa 2 - 100 pontos em 25 segundos
- Tarefa 3 - 0 pontos em 10 segundos
- Tarefa 4 - 100 pontos em 30 segundos
- Tarefa 5 - 100 pontos em 10 segundos

Pontuações gerais da turma
- Tarefa 1 - 100 pontos em 20 segundos
- Tarefa 2 - 100 pontos em 25 segundos
- Tarefa 3 - 0 pontos em 10 segundos
- Tarefa 4 - 100 pontos em 30 segundos
- Tarefa 5 - 100 pontos em 10 segundos

Análise das pontuações do aluno

Análise das pontuações da turma do aluno

Figure 22 - Student score analysis screen model

In this page, the student will be able to compare the score and time that was achieved in any task of the level that was selected against the ones that were achieved by other students on the student's class. There will also be two graphs similar to the ones that were presented in the previous pages that show the student's performance on the different areas of dyscalculia that the select level focuses on.

With all the screens relative to the student discussed, the teacher's sections of the web application are explained next, starting with the initial screen the teacher is presented (Figure 23):

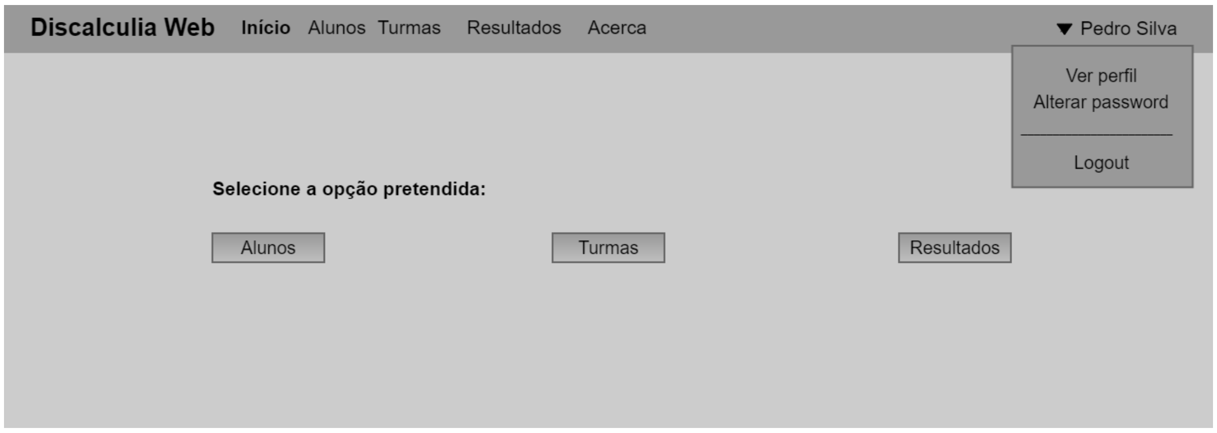


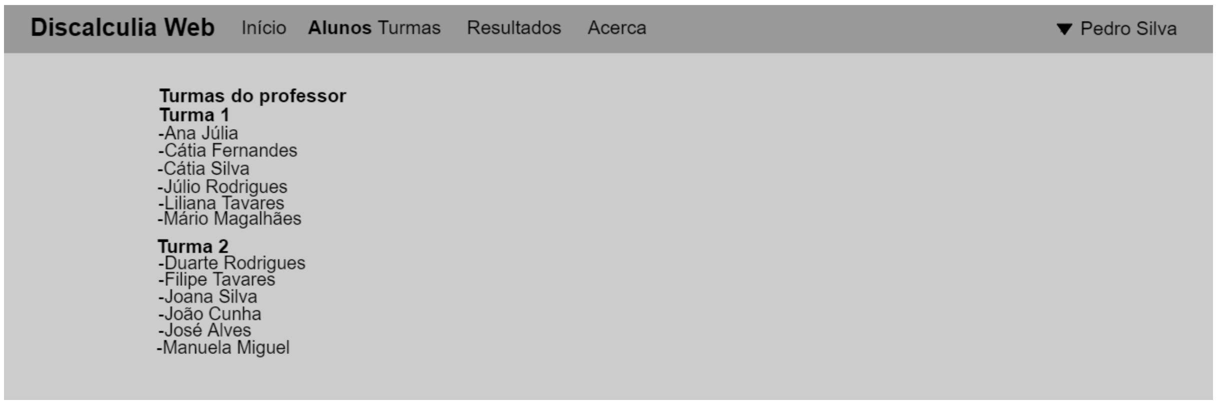
Figure 23 - Teacher initial screen model

The teacher's initial screen is not like the student's, since there is more information to be had. As it is possible to be seen the user that logs in as a teacher as three options at his disposal:

- The option to access all the students a teacher is responsible for;
- The option to see and analyze any class taught by the teacher;
- The option to access a results page that analyzes all the classes taught by the teacher.

These options are also accessible at the top of the screen by using the *Alunos*, *Turmas* and *Resultados* options, respectively.

By accessing the *Alunos* button or the *Alunos* option at the top of the screen, the teacher is shown the following screen (Figure 24):



Turmas do professor
Turma 1
-Ana Júlia
-Cátia Fernandes
-Cátia Silva
-Júlio Rodrigues
-Liliana Tavares
-Mário Magalhães

Turma 2
-Duarte Rodrigues
-Filipe Tavares
-Joana Silva
-João Cunha
-José Alves
-Manuela Miguel

Figure 24 - Teacher's students screen model

This screen shows all the students that are taught by the logged in teacher sorted by class. Each student is clickable and allow the teacher to analyze any student that he is responsible for. That analysis is done by showing the teacher the same screens that the student has access to, screens that are shown at the start of this model's section.
By clicking the second button of the first screen (Figure 23), *Turmas*, or the option with the same name on the top of screen, the user has access to the following screen (Figure 25):

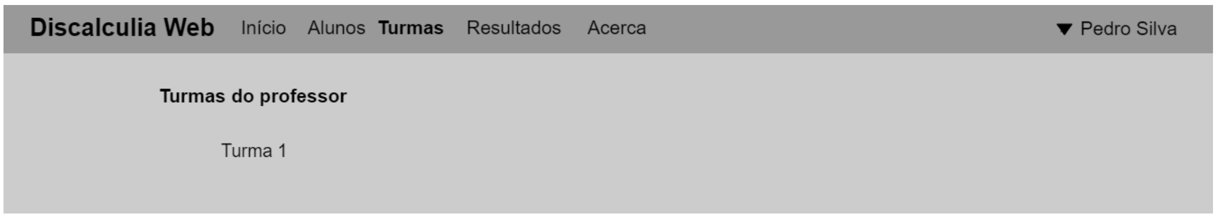


Figure 25 - Teacher's classes screen model

The teacher can see all the classes he teaches and, by clicking any class, he can analyze its performance and compare it the results achieved by other classes.
As such, the following screen (Figure 26) is shown, for the selected class:

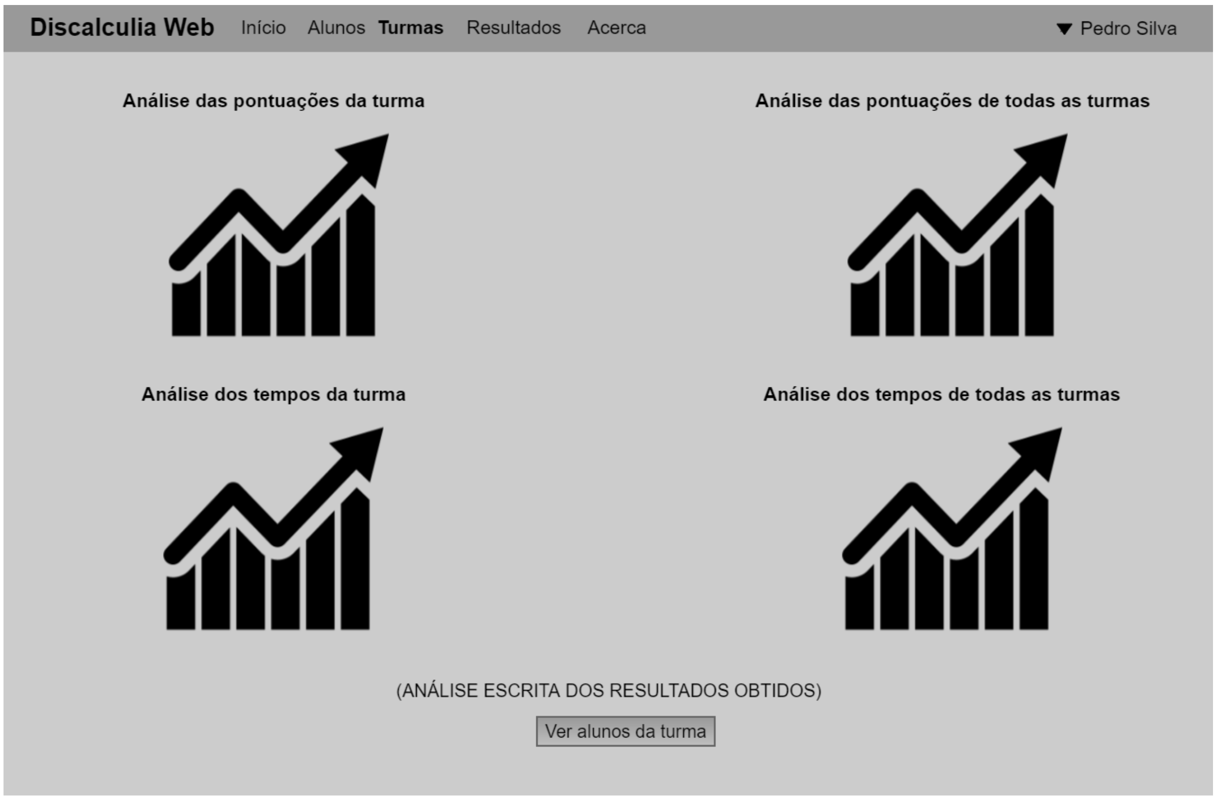
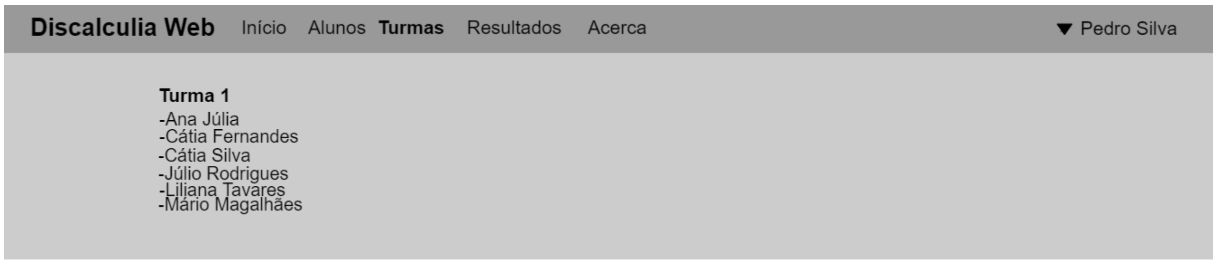

Discalculia Web
Início
Alunos
Turmas
Resultados
Acerca
▼ Pedro Silva
Análise das pontuações da turma
Análise das pontuações de todas as turmas
Análise dos tempos da turma
Análise dos tempos de todas as turmas
(ANÁLISE ESCRITA DOS RESULTADOS OBTIDOS)
Ver alunos da turma

Figure 26 - Teacher's class analysis screen model

This screen was designed similarly to the student's general analysis screen but, instead of being directed towards a single student, it focuses on a class. It shows four different graphs:

- A graph that shows the scores achieved by the students of the selected class on different areas of dyscalculia;
- A graph that shows the scores achieved by the all the students on different areas of dyscalculia, so it is possible to analyze the class's performance in the context of all the students;
- A graph that shows the mean time it took the students of the selected class to answer the different tasks directed towards the different areas of dyscalculia;
- A graph that shows the mean time it took all the students to answer the tasks directed towards the different areas of dyscalculia, in order to better understand the performance that the class had on said tasks.

On the bottom of the page there also is an analysis that explains the performance had by the class, as well as a button which lead to a page that allow the teacher to see all the students of the selected class. That page is as follows (Figure 27):



Turma 1
-Ana Júlia
-Cátia Fernandes
-Cátia Silva
-Júlio Rodrigues
-Liliana Tavares
-Mário Magalhães

Figure 27 - Teacher's class students screen model

This page is very similar to the page that displays all the students that are taught by the teacher (Figure 24) by, in this case, only the students of the select class is shown. Each student is also clickable which makes it possible for the teacher to have access to all the information the selected student has access to by displaying the same screens that the student has access to.

The last screen the teacher has access to (Figure 28), accessible by clicking on the third button of the first screen (Figure 23), *Resultados*, or the option at the top of the screen with the same name, is the following:

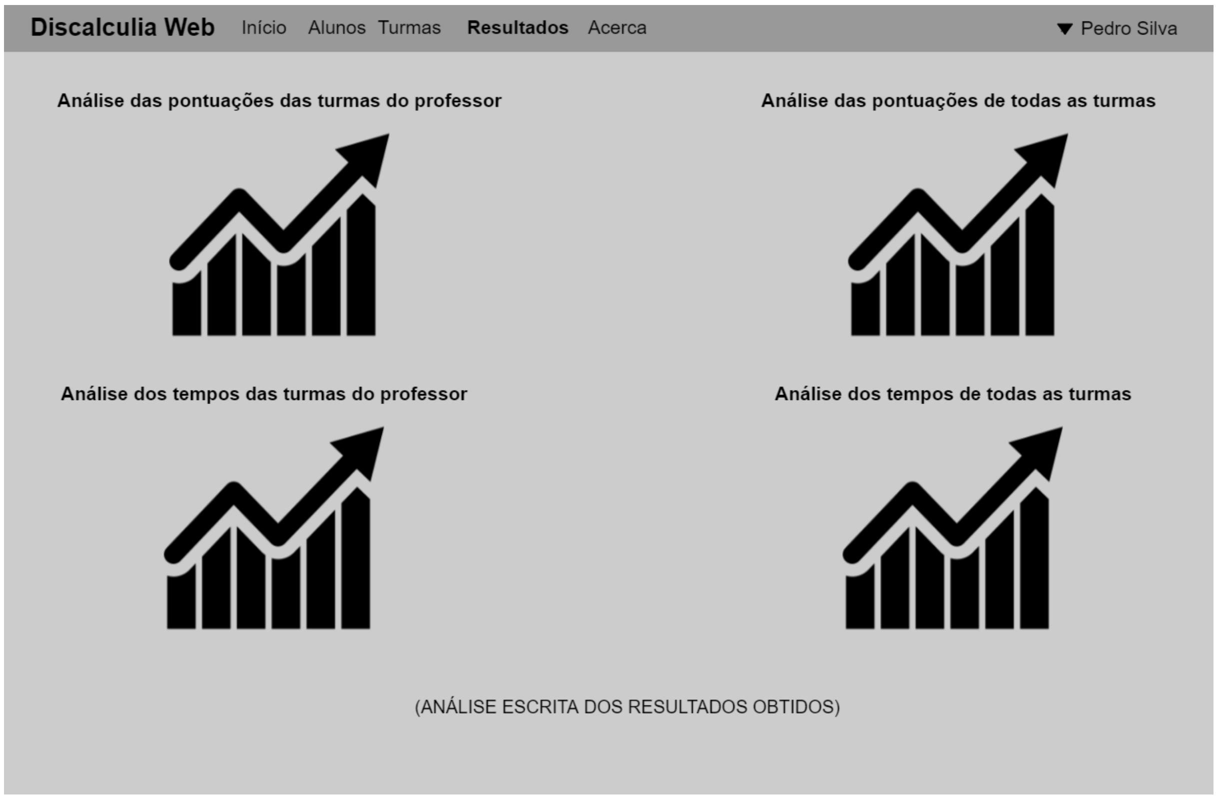

Discalculia Web
Início
Alunos
Turmas
Resultados
Acerca
▼ Pedro Silva
Análise das pontuações das turmas do professor
Análise das pontuações de todas as turmas
Análise dos tempos das turmas do professor
Análise dos tempos de todas as turmas
(ANÁLISE ESCRITA DOS RESULTADOS OBTIDOS)

Figure 28 – Teacher’s classes general analysis screen model

This screen is just like the screen that analyzes a class’s performance on different tasks (Figure 26) by comparing it with all the student’s scores and time on different areas of dyscalculia. The main difference is that the screen is directed to all classes that are taught by the logged in teacher, in order to understand, as best as possible, their overall performance.

The next section of the modeling is directed to the expert's side of the web application, starting with the screen the expert is shown when logging in (Figure 29):

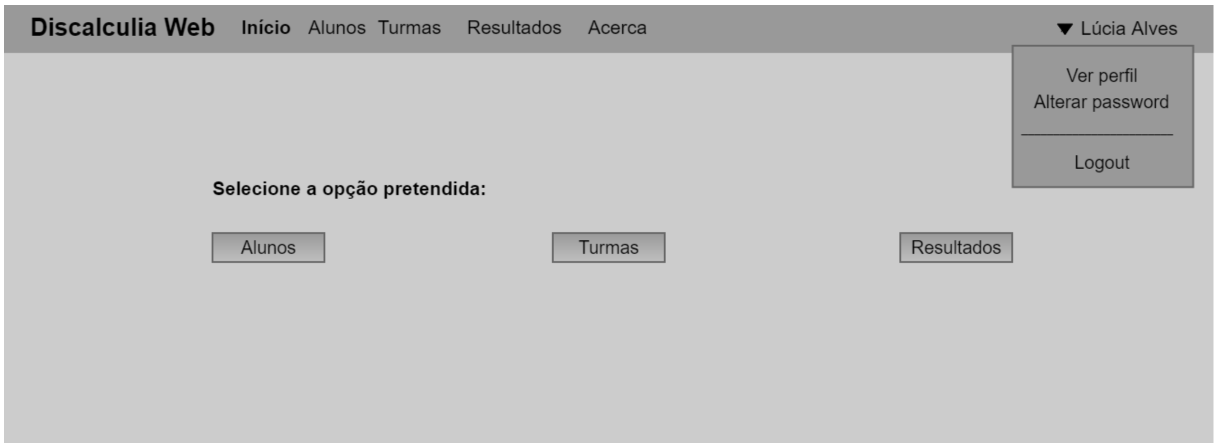
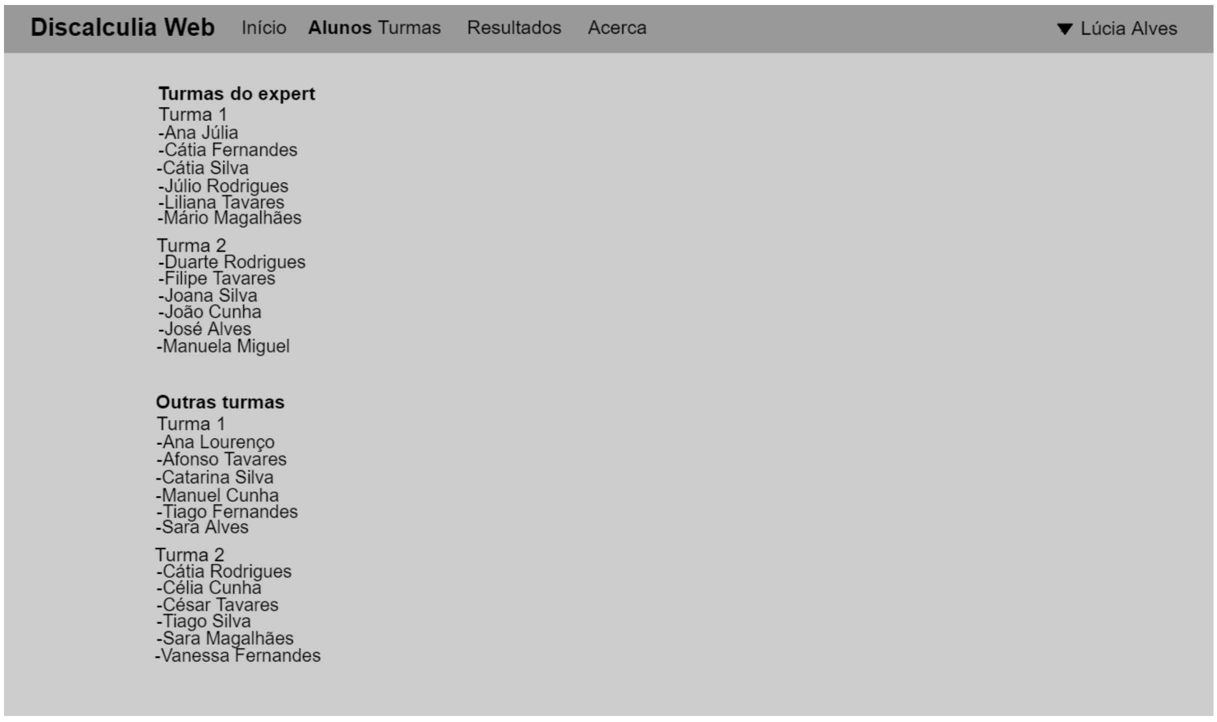


Figure 29 - Expert initial screen model

This screen is just like the teacher's initial screen (Figure 23) with the same options. All of these sections are very similar to the teacher's with some very minor differences that are going to be discussed as it is fit to do so.

By selecting the *Alunos* option either by clicking the first button of the initial screen (Figure 29) or by clicking it at the top of the screen, the user is shown the following page (Figure 30):



Turmas do expert
Turma 1
-Ana Júlia
-Cátia Fernandes
-Cátia Silva
-Júlio Rodrigues
-Liliana Tavares
-Mário Magalhães

Turma 2
-Duarte Rodrigues
-Filipe Tavares
-Joana Silva
-João Cunha
-José Alves
-Manuela Miguel

Outras turmas
Turma 1
-Ana Lourenço
-Afonso Tavares
-Catarina Silva
-Manuel Cunha
-Tiago Fernandes
-Sara Alves

Turma 2
-Cátia Rodrigues
-Célia Cunha
-César Tavares
-Tiago Silva
-Sara Magalhães
-Vanessa Fernandes

Figure 30 - Expert's students screen model

This page, again, is very similar to the teacher's students screen (Figure 24). The main difference is the classes that the expert has access to. While the teacher only has access to students of his classes, the expert has access to all students, divided into two sections, students from the classes that the expert is responsible for and another one containing all the others.
Each student is clickable and leads to a section of the web application just like the one that the student that was selected has access to, in order for the expert to be able to analyze all the information available to the student.
The following screen (Figure 31) is reachable by clicking the *Turmas* button on the first page the expert has access to (Figure 29) or by selecting the option on top of the screen with the same name:

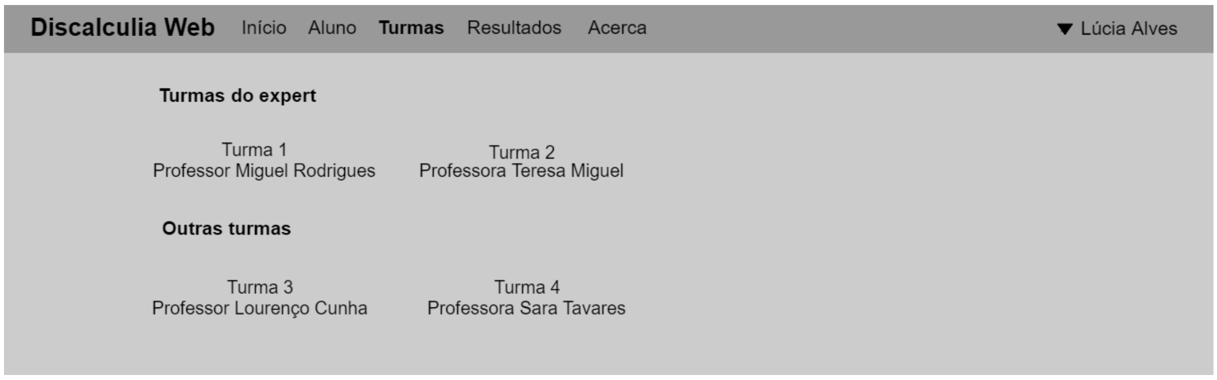


Figure 31 - Expert's classes screen model

This page contains all classes that are stored in the database and are divided into two sections, the classes that the expert that is logged in is responsible for and all the other classes. Each class is clickable and allows the user to analyze it.

That analysis can be seen by accessing the following page (Figure 32):

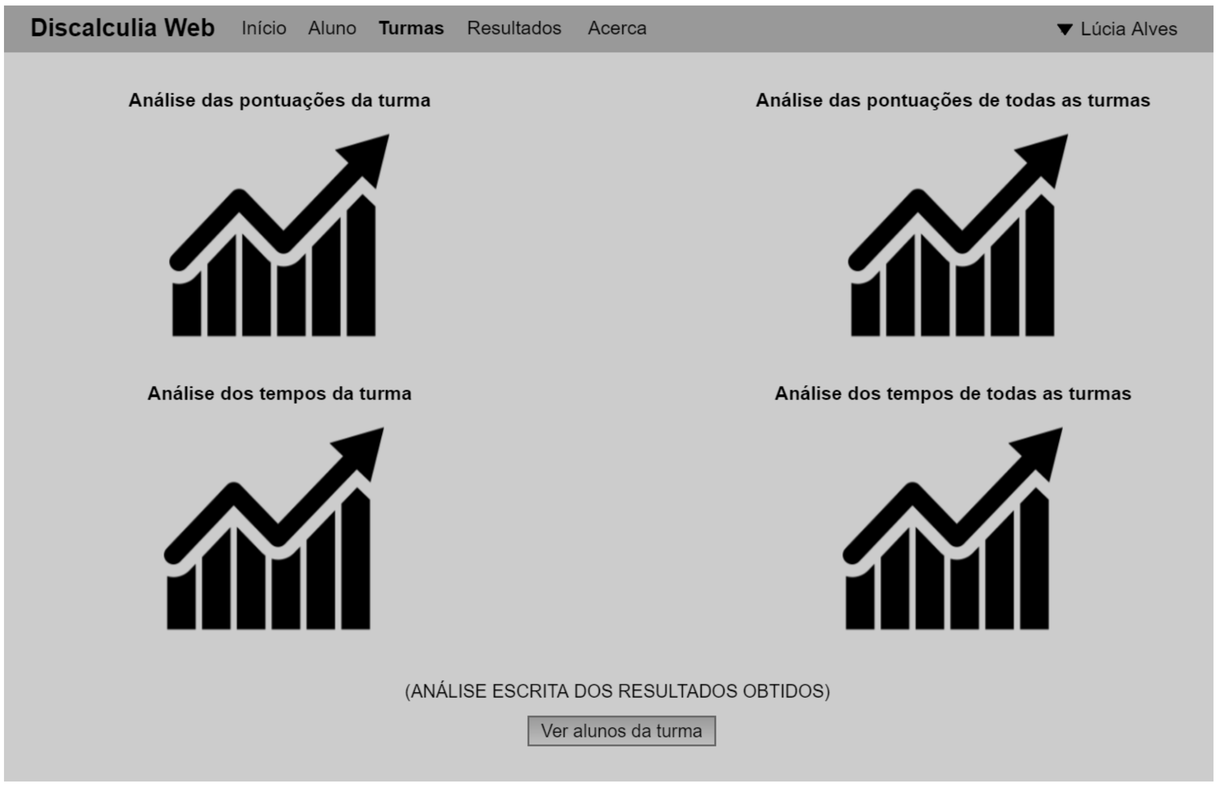
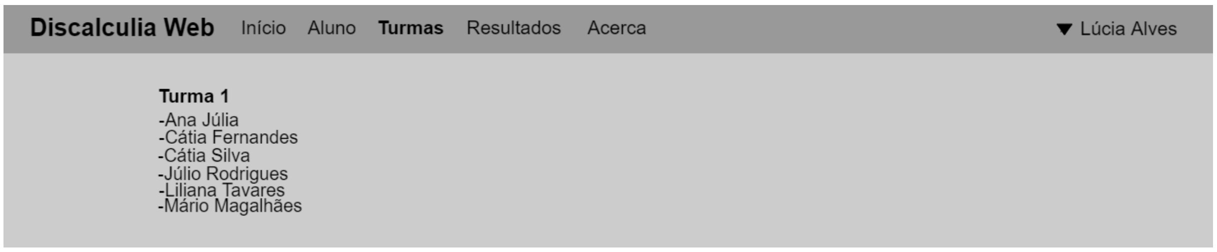

Discalculia Web
Início
Aluno
Turmas
Resultados
Acerca
▼ Lúcia Alves
Análise das pontuações da turma
Análise das pontuações de todas as turmas
Análise dos tempos da turma
Análise dos tempos de todas as turmas
(ANÁLISE ESCRITA DOS RESULTADOS OBTIDOS)
Ver alunos da turma

Figure 32 - Expert class analysis screen model

This page is just like the teacher's class analysis screen (Figure 26) and shows four different graphs that analyze the scores and time spent solving different tasks directed towards several areas of dyscalculia by the students of the selected class and compare them with the same scores and times that were achieved by all the students.

On the bottom of the page there also is a button that allows the user to see all the students of the selected class. Said button grants the user access to the following screen (Figure 33):



Turma 1
-Ana Júlia
-Cátia Fernandes
-Cátia Silva
-Júlio Rodrigues
-Liliana Tavares
-Mário Magalhães

Figure 33 - Expert class' students screen model

This screen displays all the students of the selected class and, just like several other screens that were already mentioned that contain students, they are clickable and give the expert access to the same screens and information the selected student has access to.

To conclude the screens that the expert can see there is the general analysis screen (Figure 34):

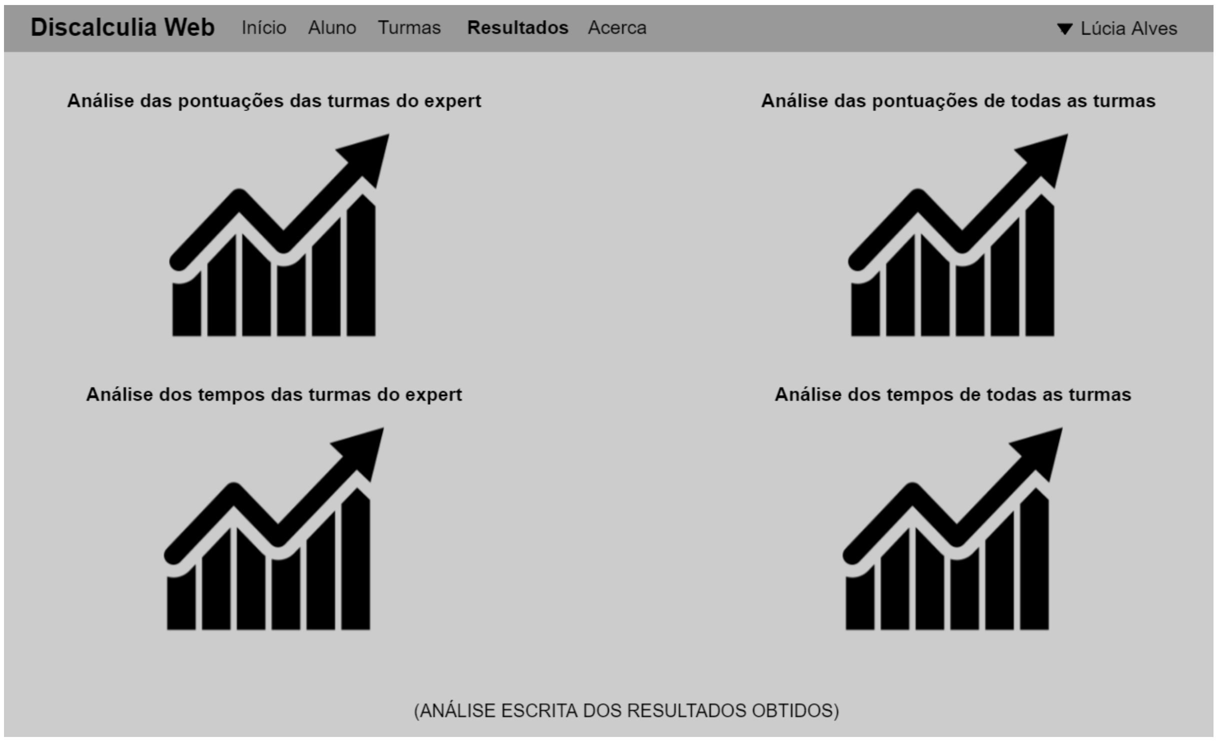

Discalculia Web
Início
Aluno
Turmas
Resultados
Acerca
▼ Lúcia Alves
Análise das pontuações das turmas do expert
Análise das pontuações de todas as turmas
Análise dos tempos das turmas do expert
Análise dos tempos de todas as turmas
(ANÁLISE ESCRITA DOS RESULTADOS OBTIDOS)

Figure 34 - Expert’s classes general analysis screen model

Just like the teacher’s classes general analysis screen (Figure 28), this screen shows the scores and time achieved by the expert’s classes on different areas of dyscalculia and compares them to the same results, this time achieved by all the students.

The final section of this model of the web application include the screens that an administrator that logs into the platform has access to, starting with the first screen that is presented (Figure 35):

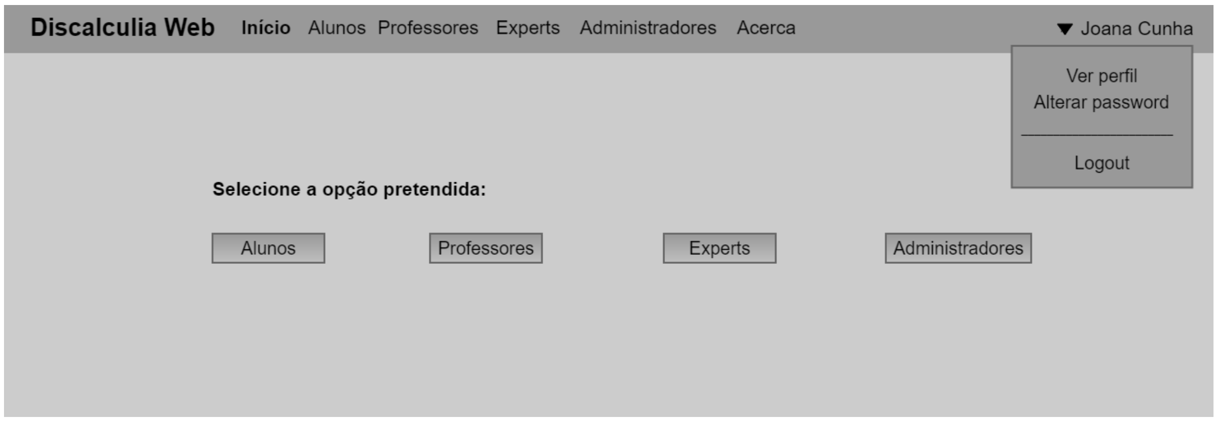


Figure 35 - Administrator initial screen model

This section is very similar to the teacher's and expert's initial screen (Figure 23 and Figure 29), but with different options available, in accordance with the administrator's functions. These options are:

- Students, that allow the administrator to access all students available in the database;
- Teachers, an option that makes it possible to see all teachers that are stored in the database;
- Experts, another option that, like the previous ones, allow access to all experts present in the platform;
- Administrators, the final option, that allows access to all administrators that exist in the database.

By selecting the first option of the initial (Figure 35) or the *Alunos* option present of the top of any page the administrator has access to, the following page is made available (Figure 36):

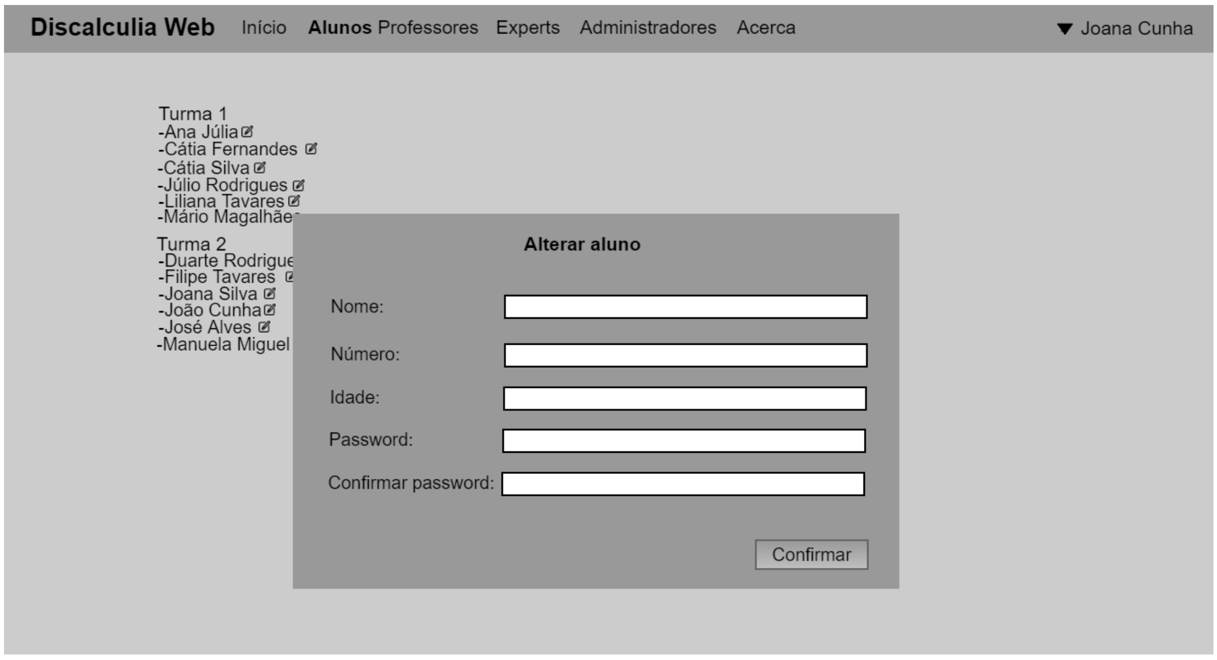


Figure 36 - Administrator students screen model

This page is very similar to any other page containing students, like the teacher's and expert's students screens (Figure 24 and Figure 30), that allow the user to see every student that is available as well as making the view over the web application that the select student has available.

The administrator has a new function that no other user type has, the ability to edit the details of any student. This ability is accessible by clicking an image next to student, an action that makes a box appear on screen that allows the administrator to change any detail, except for the identification code of the student, as mentioned previously.

The next screen (Figure 37) is accessible by selecting the second button of the initial screen that is shown to the administrator (Figure 35) or by clicking on the *Professores* option present at the top of the screen:

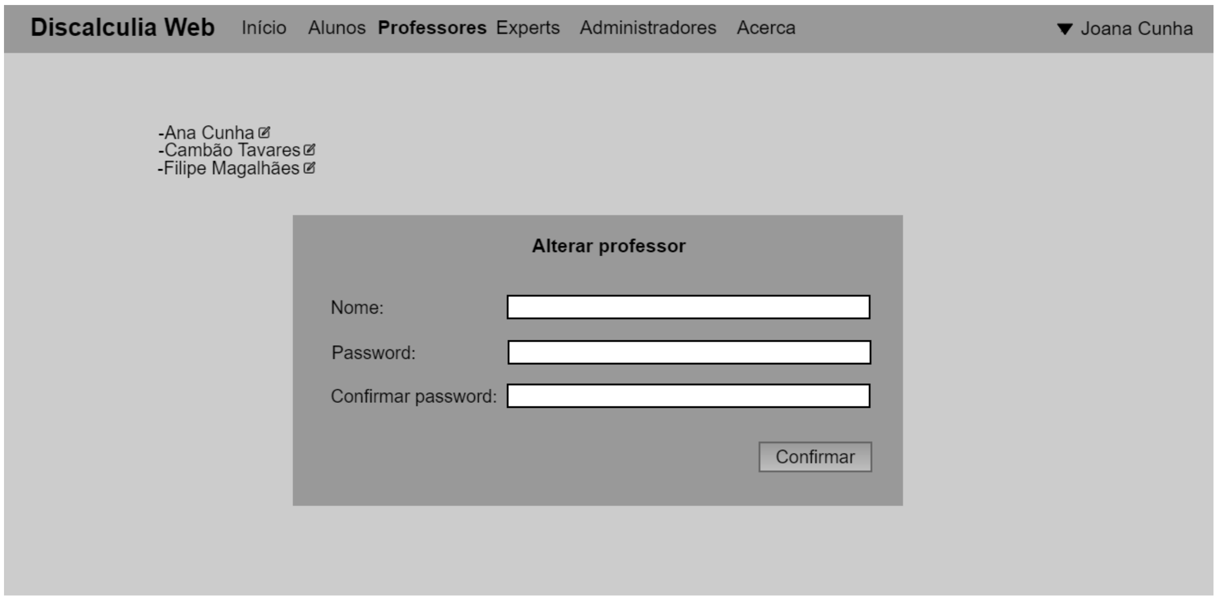


Figure 37 - Administrator teachers screen model

In this page the administrator can see all teachers that exist in the database and access the view that each one has over the web application by clicking on their name. Just like the previous screen, that is focused towards students (Figure 36), there is an image present next to the teacher that makes it possible for a pop-up to appear on screen in order to change the details of the selected teacher as easily as possible.

In order to access the next screen (Figure 38), the administrator has to select the *Experts* option on the initial screen (Figure 35) or the option with the same name on top of the screen:

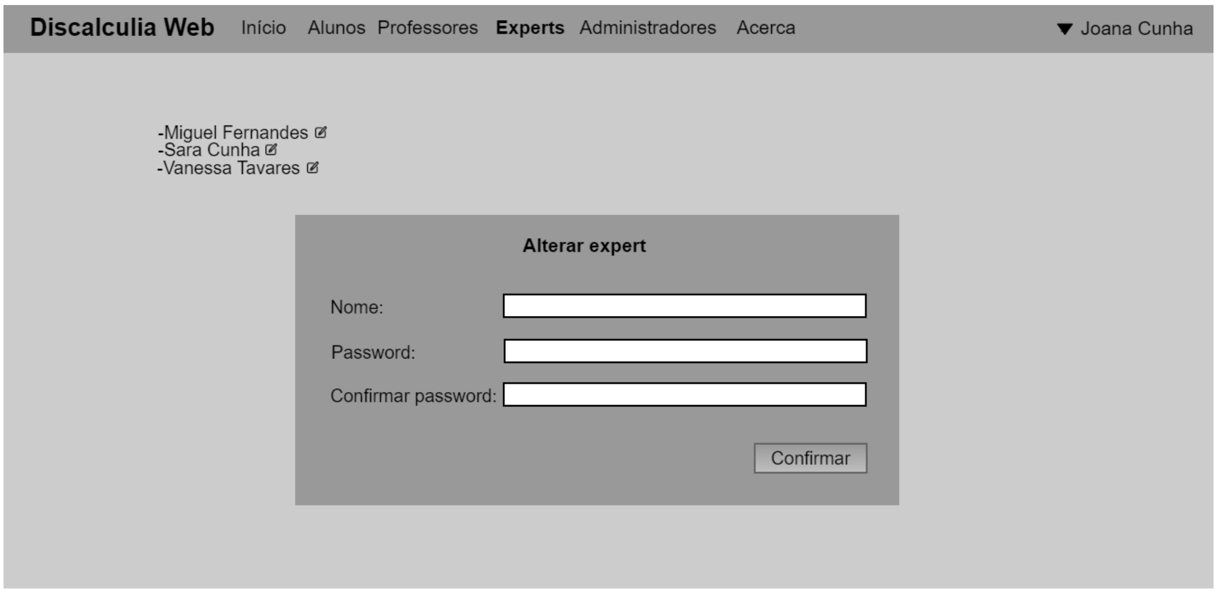


Figure 38 - Administrator experts screen model

This screen is just like the previous one but directed towards experts. It displays all experts present of the platform and grants the administrator access to the view the selected expert has over the web application. Again, next to the names of the experts, there is an image that allows the edition of the details of any expert, by showing the pop-up present of the center of the screen that is shown in the previous screen (Figure 38).

The final screen of the administrator section and of the model itself, accessible by clicking on the *Administradores* option on top of the screen of by clicking the button with the same name on the initial screen (Figure 35), is the following (Figure 39):

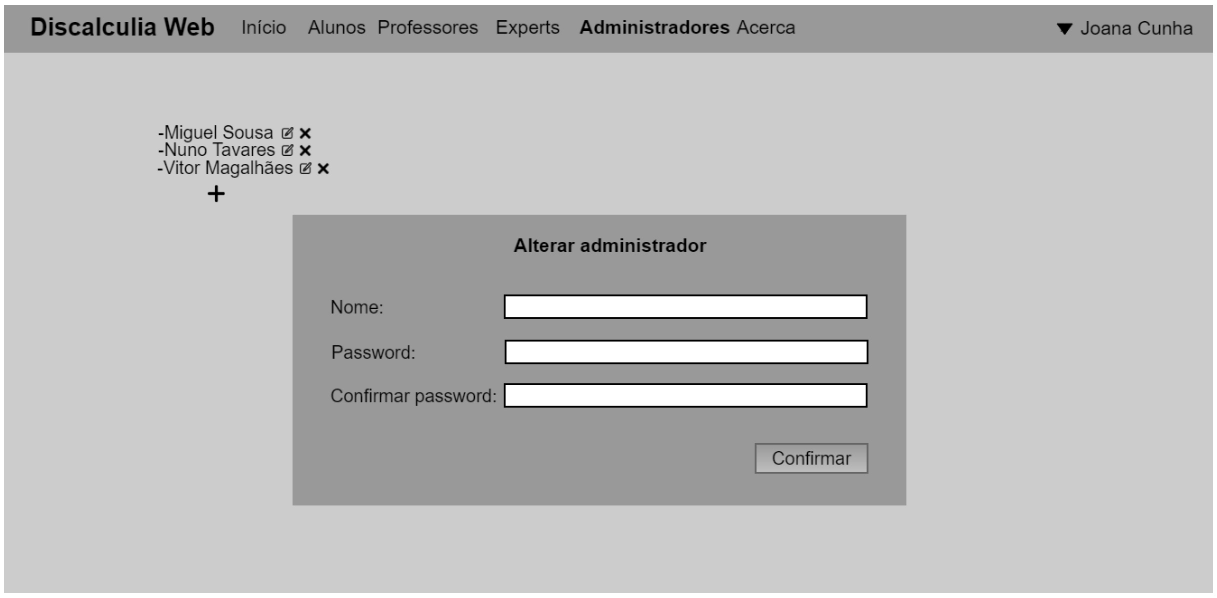


Figure 39 - Administrator administrators screen model

This screen is very similar to the last screens. The main differences consist of the fact that each administrator is not clickable, not granting access to the view that administrator has, the fact that each administrator is removable, an action made possible by clicking the cross image next to its name, and the ability to add administrator by clicking the plus image which opens a pop-up with the same fields present in the pop-up present in the center of the screen that is being discussed. It is also possible to change the details of any administrator, the action that is being displayed, by also clicking on the image that represents that action next to the name of the one that is to be altered.

Since all the modeling component of the development process of the web application was done every condition was met to start effectively building the application itself, a process that is described and discussed in the next section.

6.6. Implementation

Since the modeling was done at this point of the project the building process was majorly based upon it, since the most complex components were planned out. The steps for this process consisted of converting the models done to Angular, the selected technology for developing the web application, with some changes or not, and doing the programming aspect of all functions that were planned.

In order to start developing the web application there was decision that had to be made at this point, the decision of which web framework to use. A web framework is a software system that offers several tools directed towards web application development that, when put together, facilitate and make the construction process more efficient and less time consuming by having several features in one package [108], [109].

The web framework that was going to be used was decided to be either Bootstrap or Materialize, two popular web frameworks with very aesthetically pleasing components which used proved to be positive in past experiences. Initially the one that was going to be used was Materialize due to the fact that is focused on a more minimalist design. But the final choice was Bootstrap since, during the testing phase of these frameworks, it was the most practical of the two and, unlike Materialize, there were no problems when using certain components together on the same page [110], [111].

Also, just like it was mentioned in the Technologies and methods section, the selected code editor was Visual Studio Code, a software program that was also very useful to test the web application during the development process together with a tool called Node.js that allows the user to run the type of code that was used, TypeScript.

As such, the final result was as follows, starting with the first page (Figure 40) that is displayed after connecting to the web application though its public internet address:

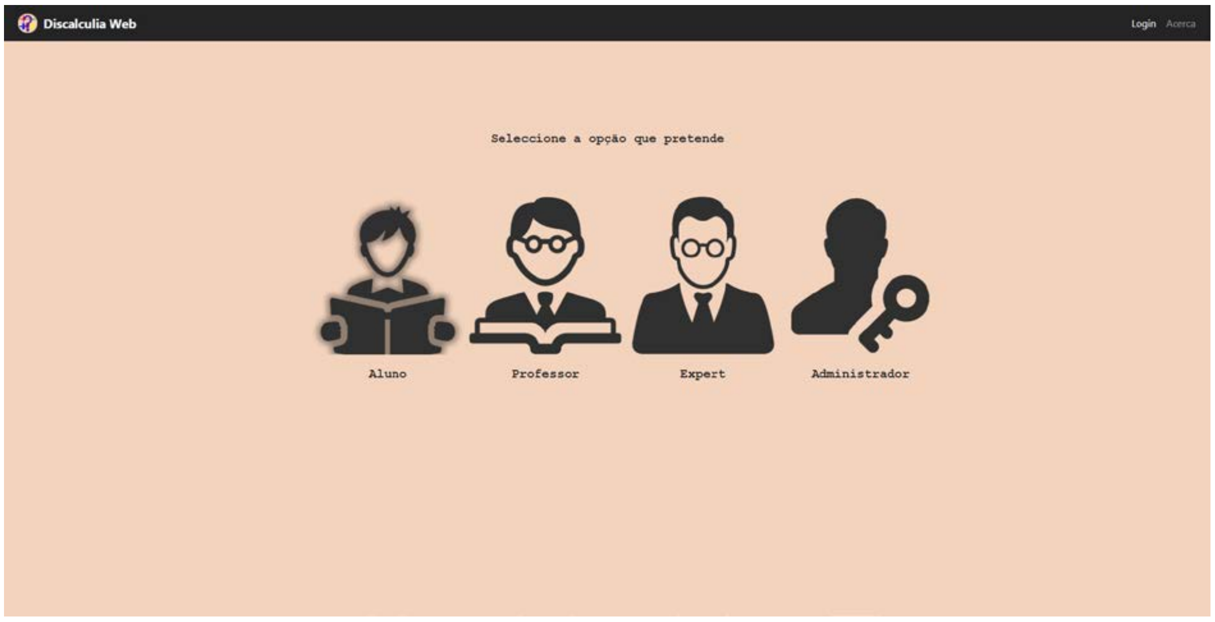


Figure 40 – Initial screen preview

On this screen, the user is presented four options (Figure 40) regarding the user type that he wishes to login as: student, teacher, expert and administrator, in that order. When the images displayed on this page are hovered over with the mouse pointer they are pushed forward and a shadow is added to them, as can be seen in the leftmost image, an effect that adds to the page's fluidity and that shows the user that each image is clickable.

On the top left of the screen there is a logo (Figure 42) that was created for the web application, according to the visual style chosen for the web application, and the name of the platform, Discalculia Web (Figure 41), which is clickable and leads to the initial screen (Figure 40).

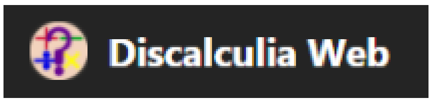


Figure 41 - Web application brand

Figure 42 - Web application logo

On the top right of the screen there are two options, *Login* and *Acerca*, as can be seen in this close-up figure (Figure 43):

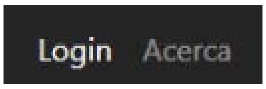
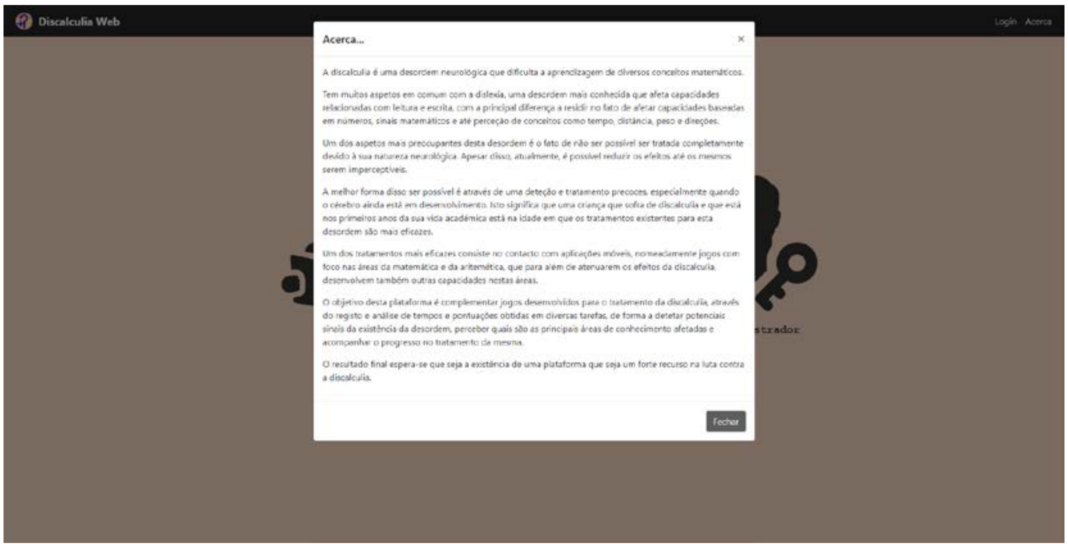


Figure 43 - Login and *Acerca* close-up

By clicking Login, just like with the platform brand, the user is redirected towards the initial screen (Figure 40). By clicking on the *Acerca* option, the user is presented a pop-up that explain the purpose of this project, as can be seen in the following figures (Figure 44 and Figure 45):



Acerca...

A discalculia é uma desordem neurológica que dificulta a aprendizagem de diversos conceitos matemáticos.

Tem muitos aspetos em comum com a dislexia, uma desordem mais conhecida que afeta capacidades relacionadas com leitura e escrita, com a principal diferença a residir no fato de afetar capacidades baseadas em números, sinais matemáticos e até perceção de conceitos como tempo, distância, peso e direções.

Um dos aspetos mais preocupantes desta desordem é o fato de não ser possível ser tratada completamente devido à sua natureza neurológica. Apesar disso, atualmente, é possível reduzir os efeitos até os mesmos serem impercetíveis.

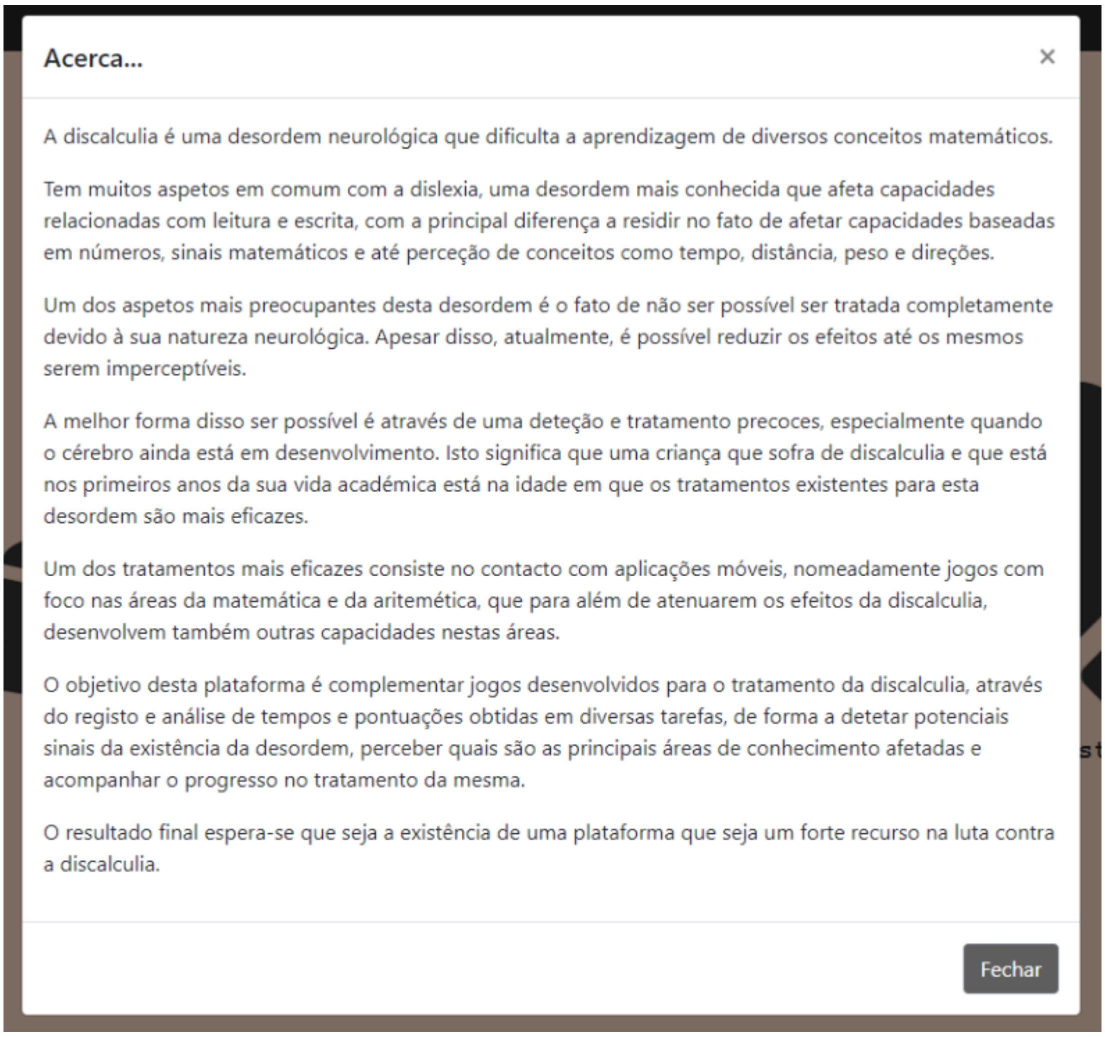
A melhor forma disso ser possível é através de uma deteção e tratamento precoces, especialmente quando o cérebro ainda está em desenvolvimento. Isto significa que uma criança que sofra de discalculia e que está nos primeiros anos da sua vida académica está na idade em que os tratamentos existentes para esta desordem são mais eficazes.

Um dos tratamentos mais eficazes consiste no contacto com aplicações móveis, nomeadamente jogos com foco nas áreas da matemática e da aritemética, que para além de atenuarem os efeitos da discalculia, desenvolvem também outras capacidades nestas áreas.

O objetivo desta plataforma é complementar jogos desenvolvidos para o tratamento da discalculia, através do registo e análise de tempos e pontuações obtidas em diversas tarefas, de forma a detetar potenciais sinais da existência da desordem, perceber quais são as principais áreas de conhecimento afetadas e acompanhar o progresso no tratamento da mesma.

O resultado final espera-se que seja a existência de uma plataforma que seja um forte recurso na luta contra a discalculia.

Figure 44 – *Acerca* screen preview


Acerca...

×

A discalculia é uma desordem neurológica que dificulta a aprendizagem de diversos conceitos matemáticos.

Tem muitos aspetos em comum com a dislexia, uma desordem mais conhecida que afeta capacidades relacionadas com leitura e escrita, com a principal diferença a residir no fato de afetar capacidades baseadas em números, sinais matemáticos e até perceção de conceitos como tempo, distância, peso e direções.

Um dos aspetos mais preocupantes desta desordem é o fato de não ser possível ser tratada completamente devido à sua natureza neurológica. Apesar disso, atualmente, é possível reduzir os efeitos até os mesmos serem imperceptíveis.

A melhor forma disso ser possível é através de uma deteção e tratamento precoces, especialmente quando o cérebro ainda está em desenvolvimento. Isto significa que uma criança que sofra de discalculia e que está nos primeiros anos da sua vida académica está na idade em que os tratamentos existentes para esta desordem são mais eficazes.

Um dos tratamentos mais eficazes consiste no contacto com aplicações móveis, nomeadamente jogos com foco nas áreas da matemática e da aritemética, que para além de atenuarem os efeitos da discalculia, desenvolvem também outras capacidades nestas áreas.

O objetivo desta plataforma é complementar jogos desenvolvidos para o tratamento da discalculia, através do registo e análise de tempos e pontuações obtidas em diversas tarefas, de forma a detetar potenciais sinais da existência da desordem, perceber quais são as principais áreas de conhecimento afetadas e acompanhar o progresso no tratamento da mesma.

O resultado final espera-se que seja a existência de uma plataforma que seja um forte recurso na luta contra a discalculia.

Fechar

Figure 45 - *Acerca* pop-up close-up (Portuguese text)

In this pop-up, the following translated text can be read:

Dyscalculia is a neurological disorder that makes the learning process of several mathematical concepts harder to learn.
It has many similarities with dyslexia, a better-known disorder that hampers abilities related to reading and writing, that is different when compared to dyscalculia, since the latter affects capabilities related to numbers, mathematical symbols and several concepts such as time, distance, weight and directions.
One of its most worrying characteristics is the fact that it cannot be completely treated due to its neurological origin. However, in the current days, it is possible to fight its symptoms to a point where they are not noticeable.
The best way to do that consists in an early detection and treatment, when the brain is still

developing. This means that a child that is starting school or on the first years of school life and suffers from dyscalculia is a prime candidate for the most effective treatment possible for this disorder.

One of the most effective treatments available consists of the contact with mobile applications, namely games that focus on mathematics and arithmetic that hamper the effects of dyscalculia and improve other capabilities on these areas.

The aim of this platform is to complement games that were developed with dyscalculia in mind, by registering and analyzing times and scores achieved in different tasks, in order to try to detect potential signs of dyscalculia, understand which areas of knowledge it affects most and accompany the progress in the treatment of this disorder.

The final result is a platform that aims to be a strong aid in fighting dyscalculia.

By selecting one of the options presented in the initial screen (Figure 40), the user is shown the following (Figure 46):

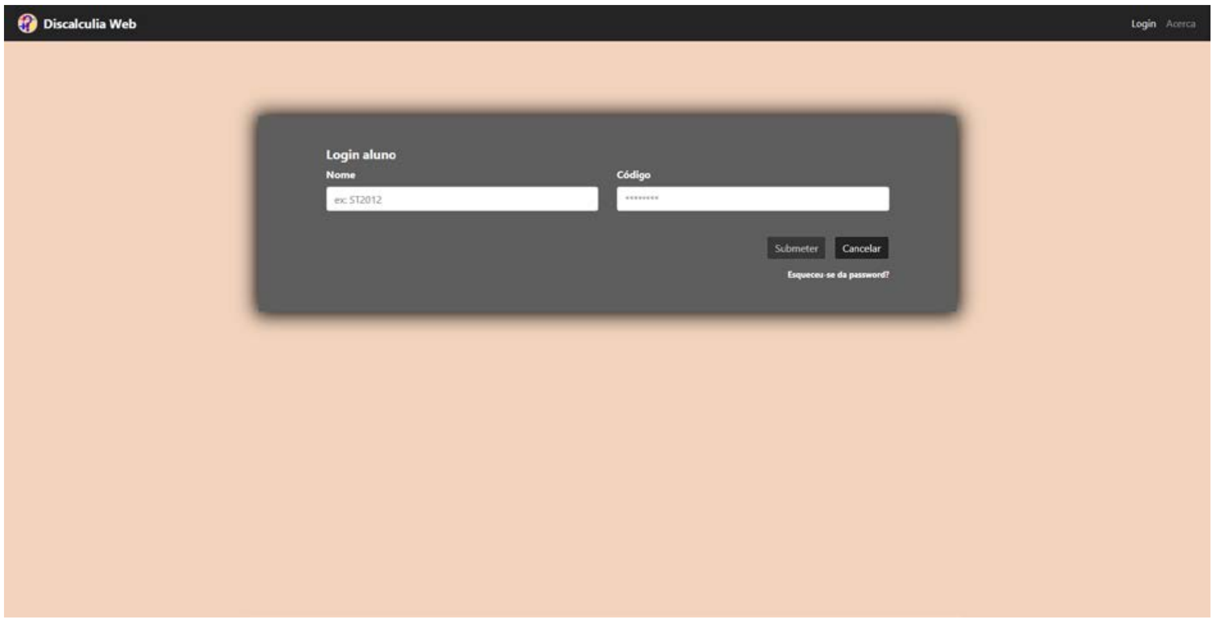
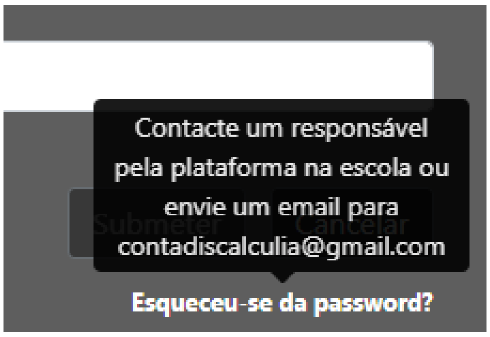


Figure 46 – Login screen preview

The previous figure shows the screen that is displayed after clicking the student option on the initial screen (Figure 40) but the other users' screens are the same as this one. The left input box asks the user for the login name and the right input box asks for the password.

Under the *Submeter* and *Cancelar* buttons, the buttons that are responsible for submitting the inserted data and for logging in to the platform, and canceling the login process and returning to the initial screen (Figure 40), there is a white piece of text that reads "*Esqueceu-se da password?*" that asks the user if the password is forgotten. By hovering over it, the following pop-up appears (Figure 47):



Contacte um responsável pela plataforma na escola ou envie um email para contadiscalculia@gmail.com

Figure 47 - Forgotten password pop-up

This pop-up tells the user the following translated message:

> *Contact a person responsible for the platform on school or send an email to contadiscalculia@gmail.com.*

The email address that is given is an actual email address that currently exists and was created for resetting any forgotten password or solve any difficulty a user can have while regarding the platform.

After logging as a student, one of the following screens is shown (Figure 48 and Figure 49):

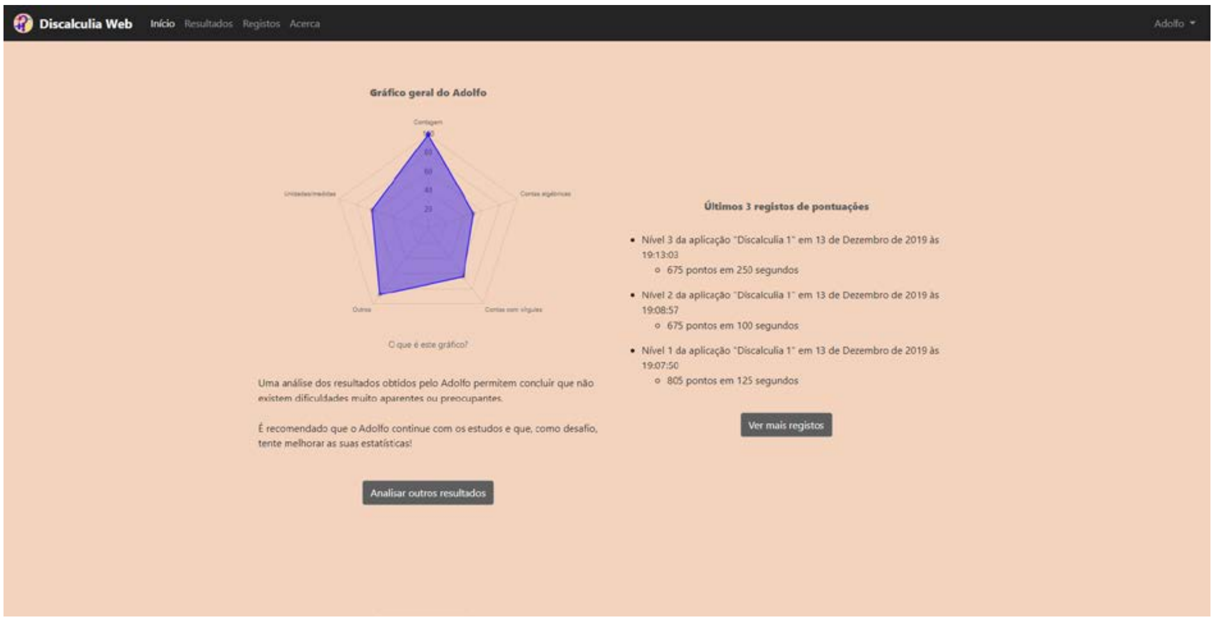


Gráfico geral do Adolfo

Uma análise dos resultados obtidos pelo Adolfo permitem concluir que não existem dificuldades muito aparentes ou preocupantes.

É recomendado que o Adolfo continue com os estudos e que, como desafio, tente melhorar as suas estatísticas!

Últimos 3 registos de pontuações

- Nível 3 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:13:03
  - 675 pontos em 250 segundos
- Nível 2 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:08:57
  - 675 pontos em 100 segundos
- Nível 1 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:07:50
  - 805 pontos em 125 segundos

Figure 48 - Male student screen preview

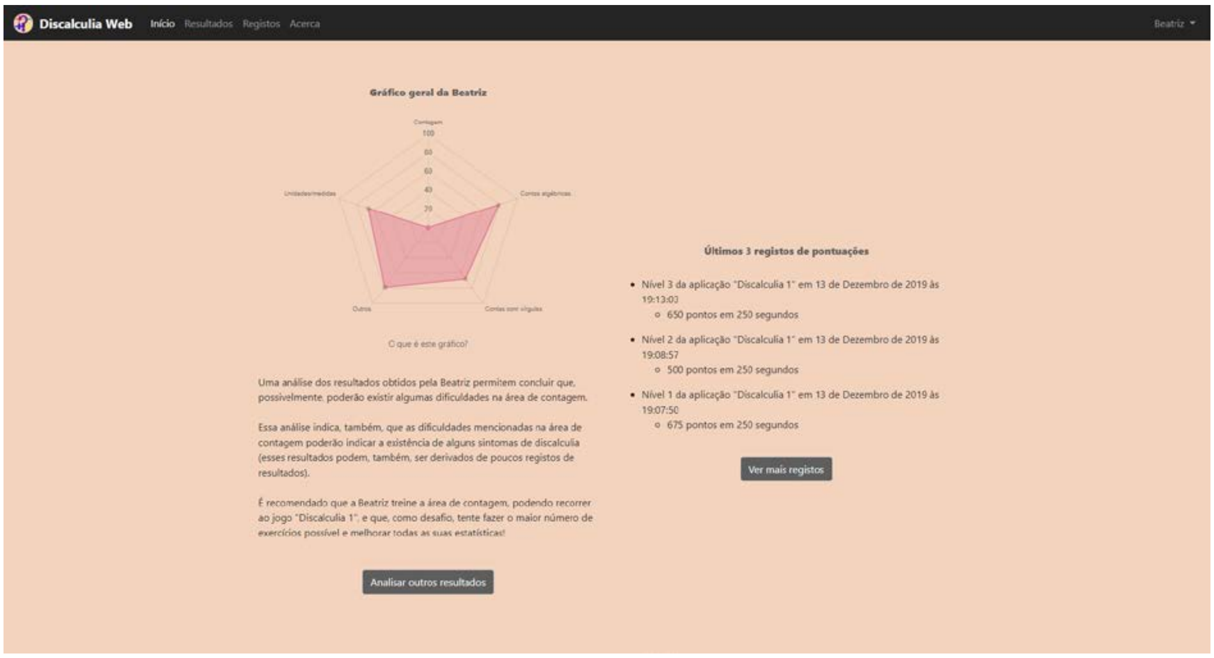


Gráfico geral da Beatriz

Uma análise dos resultados obtidos pela Beatriz permitem concluir que, possivelmente, poderão existir algumas dificuldades na área de contagem.

Essa análise indica, também, que as dificuldades mencionadas na área de contagem poderão indicar a existência de alguns sintomas de discalculia (esses resultados podem, também, ser derivados de poucos registos de resultados).

É recomendado que a Beatriz treine a área de contagem, podendo recorrer ao jogo "Discalculia 1", e que, como desafio, tente fazer o maior número de exercícios possível e melhorar todas as suas estatísticas!

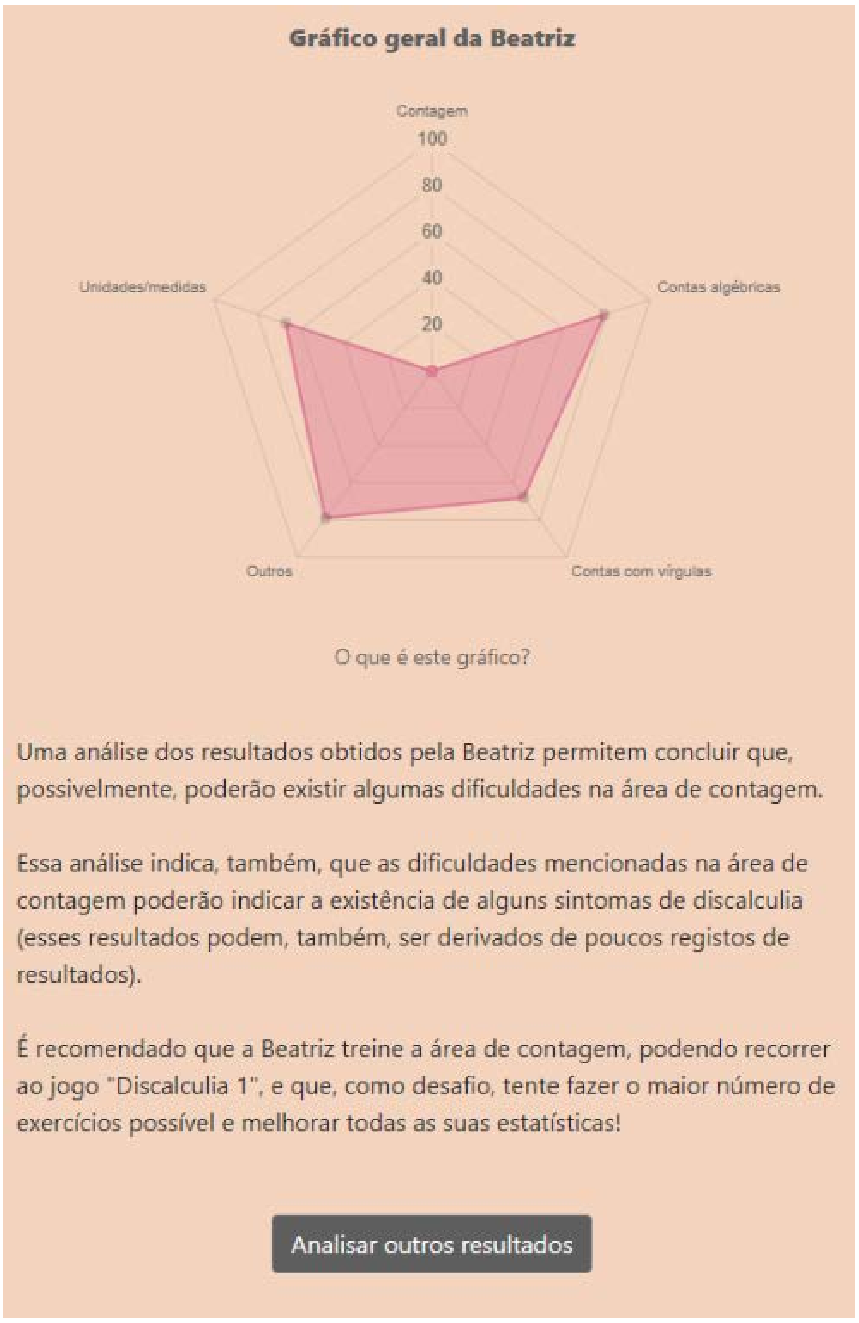
Últimos 3 registos de pontuações

- Nível 3 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:13:03
  - 650 pontos em 250 segundos
- Nível 2 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:08:57
  - 500 pontos em 250 segundos
- Nível 1 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:07:50
  - 675 pontos em 250 segundos

Figure 49 - Female student screen preview

These screens are shown according to the student's gender, the first (Figure 48) is shown to male students and the second (Figure 49) is shown to female students, and the main difference is the color chosen for the graph that is presented.

On the left side of the screen (Figure 50), there is a type of graph called radar chart, as mentioned in the requirements' analysis section of the web application, that shows the score percentage regarding the tasks of the different areas of dyscalculia the student was tested on. There also is a written analysis of these results that talks about potential areas of knowledge where there might be dyscalculia symptoms, areas that the student has difficulties in, and game recommendations based on the student's difficulties, and a button, *Analisar outros resultados*, which leads the user to a page with a more detailed analysis of the student's performance.



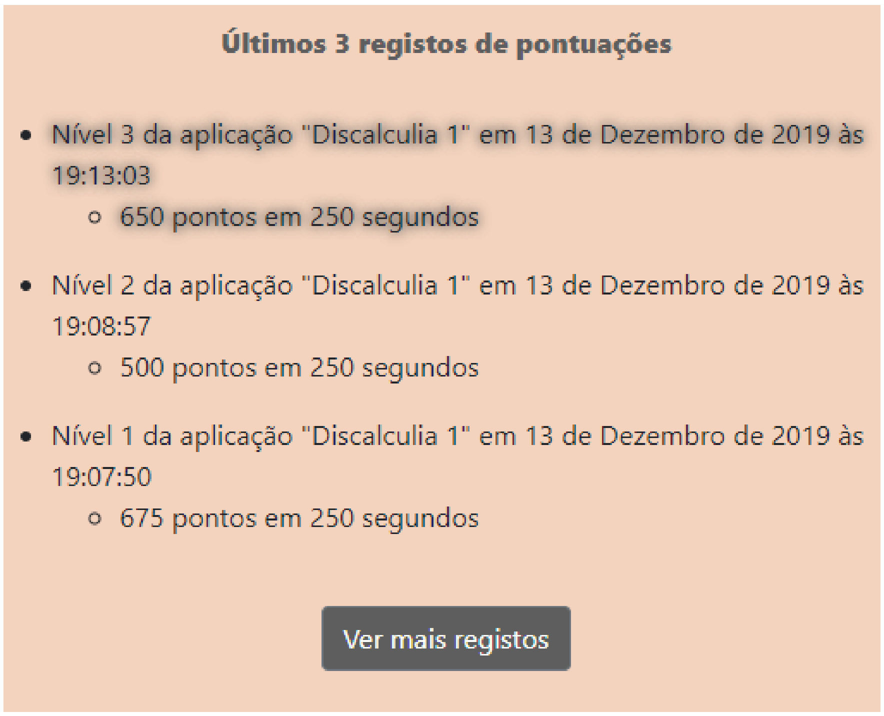
Uma análise dos resultados obtidos pela Beatriz permitem concluir que, possivelmente, poderão existir algumas dificuldades na área de contagem.

Essa análise indica, também, que as dificuldades mencionadas na área de contagem poderão indicar a existência de alguns sintomas de discalculia (esses resultados podem, também, ser derivados de poucos registos de resultados).

É recomendado que a Beatriz treine a área de contagem, podendo recorrer ao jogo "Discalculia 1", e que, como desafio, tente fazer o maior número de exercícios possível e melhorar todas as suas estatísticas!

Figure 50 - Student screen left side close-up

On the right side of the screen (Figure 51), the user is presented the latest three results that were achieved by the student. Each of these results shows the game and level that the score was achieved in, as well as the date it was registered, the score and the time. By hovering with the mouse pointer over any result there is a shadow that appears, shown in the first result of Figure 51, which indicates the user that it is clickable. By clicking in any result, the user is redirected to a page that analyzes the score that was achieved. There also is a button, *Ver mais registos*, that presents the user with a page that contains every result achieved by the student.


Últimos 3 registos de pontuações

- Nível 3 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:13:03
  - 650 pontos em 250 segundos
- Nível 2 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:08:57
  - 500 pontos em 250 segundos
- Nível 1 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:07:50
  - 675 pontos em 250 segundos

Figure 51 - Student screen right side close-up

On the top left of the screen the user is presented the web application brand as well as different options (Figure 52):

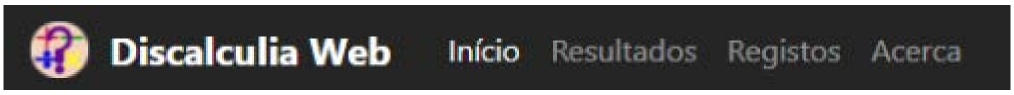


Figure 52 - Student screen top left close-up

These options lead the user to different pages: *Início* leads the user to the initial student page (Figure 48), *Resultados* leads to the page where the student's performance is analyzed, just like the *Analisar outros* *resultados* button in the left side of the screen (Figure 50), *Registos* which leads the user to the page containing all of the student's scores, just like the Ver mais registos button on the right side of the screen (Figure 51). *Acerca* makes the pop-up that explains the project appear on screen (Figure 45).

On the top right side of the screen the user is presented the name of the student that he is logged in as and is shown two further options if he clicks on the name (Figure 53):

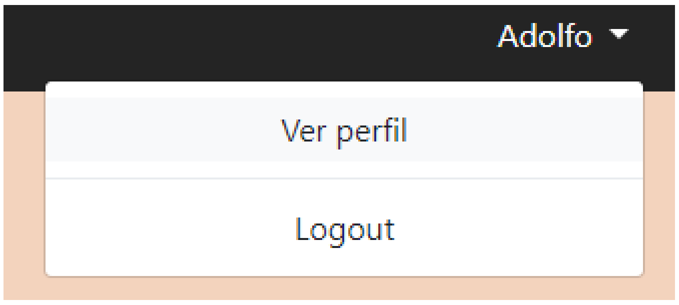
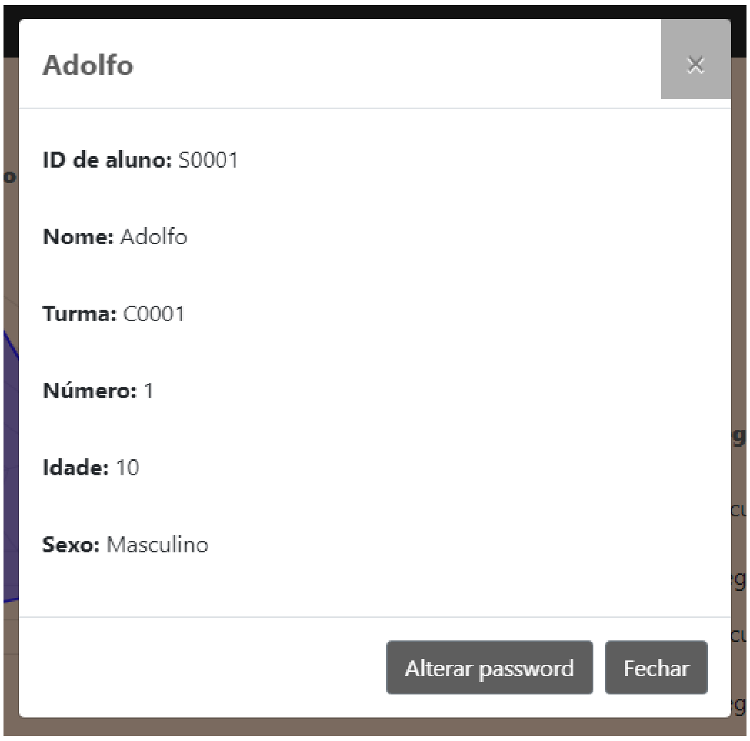


Figure 53 - Student screen top right side close-up

The second option, *Logout*, allows the user to logout of the student's account and return to the initial screen (Figure 40). The first option, *Ver perfil*, allows the user to the see the student's profile and the information that is stored regarding the student's identity (Figure 54):



Adolfo

ID de aluno: S0001

Nome: Adolfo

Turma: C0001

Número: 1

Idade: 10

Sexo: Masculino

Figure 54 - Student profile close-up

Here, the user can see the student's identification code, name, class identification code, class number, age and gender. On the bottom of the pop-up there are two options. The second option, *Fechar,* allows the user to close the pop-up, and the *Alterar password* option allows the user to change the student's password by showing a new pop-up (Figure 55):

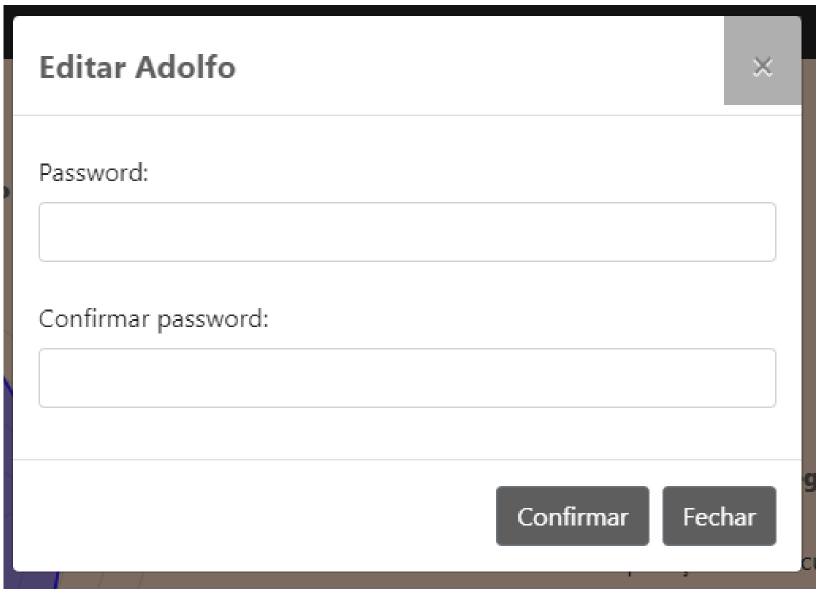
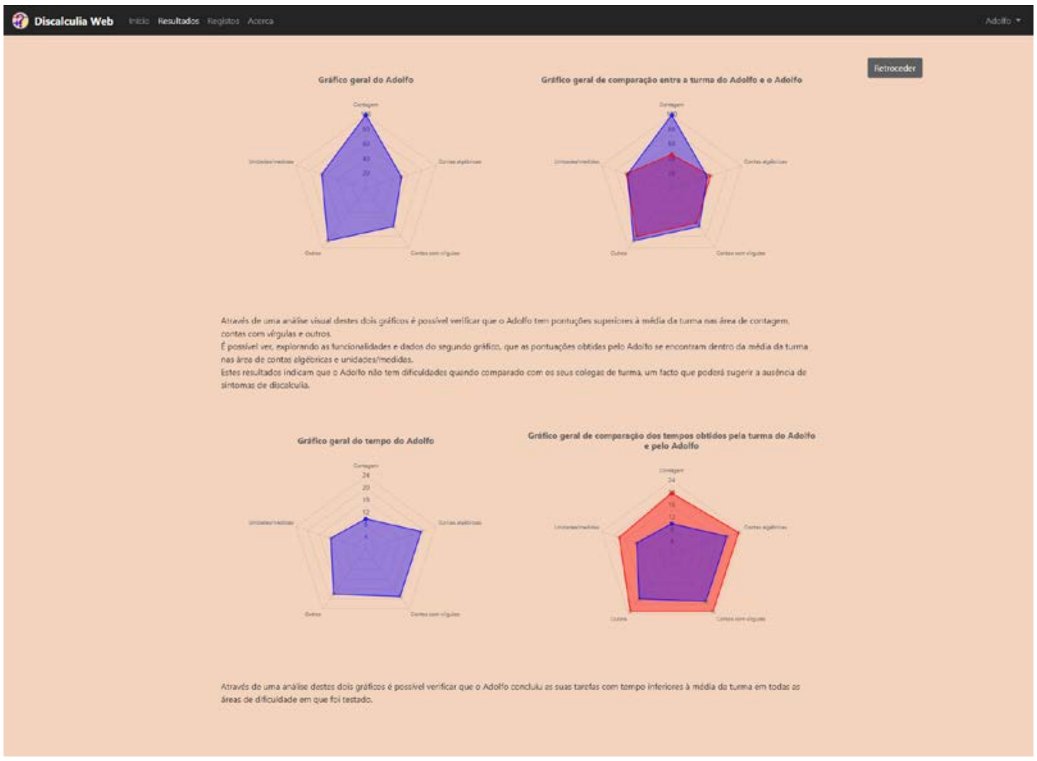


Figure 55 - Student password edition pop-up close-up

This pop-up asks the user to insert the new password in the first field and repeat it on the second and, if they match, the student's password is changed.

By clicking in the *Resultados* option on the top left of the screen (Figure 52) or by clicking the *Analisar outros resultados* button on the left side of the student's initial page (Figure 50), the user is presented the following page (Figure 56):


Discalculia Web
Início
Resultados
Registos
Acerca
Adolfo
Retroceder
Gráfico geral do Adolfo
Gráfico geral de comparação entre a turma do Adolfo e o Adolfo
Contagem
Contas algébricas
Contas com vírgulas
Outros
Unidades/medidas
Gráfico geral do tempo do Adolfo
Gráfico geral de comparação dos tempos obtidos pela turma do Adolfo e pelo Adolfo

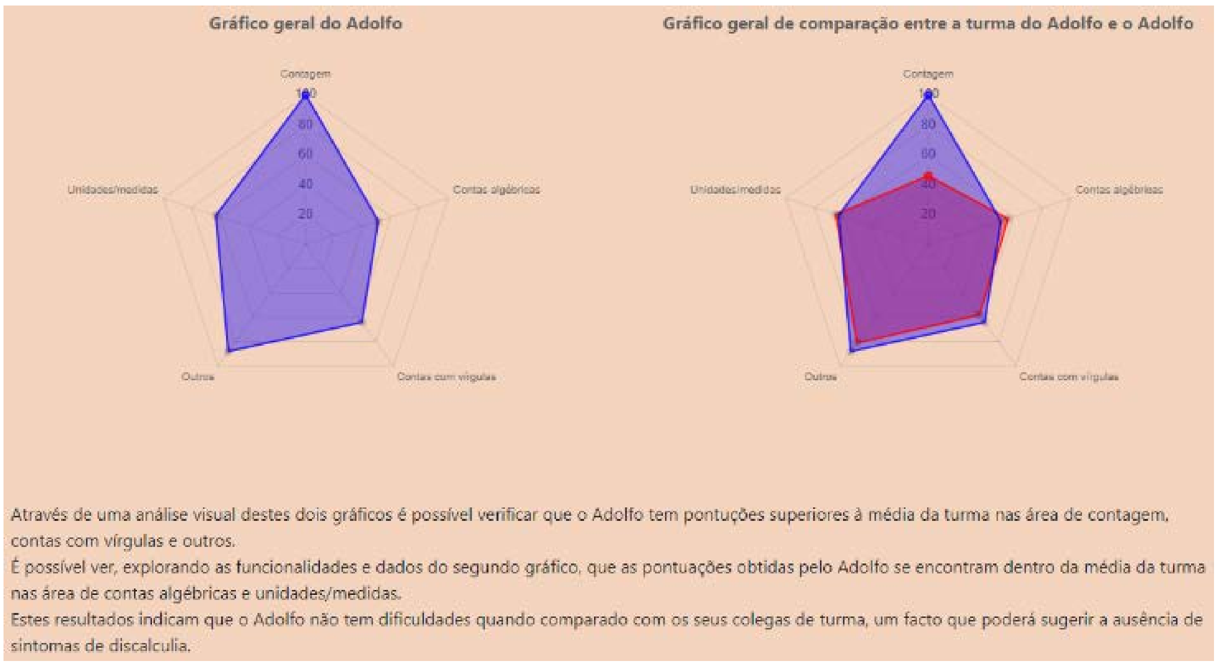
Através de uma análise visual destes dois gráficos é possível verificar que o Adolfo tem pontuções superiores à média da turma nas área de contagem, contas com vírgulas e outros.
É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pelo Adolfo se encontram dentro da média da turma nas área de contas algébricas e unidades/medidas.
Estes resultados indicam que o Adolfo não tem dificuldades quando comparado com os seus colegas de turma, um facto que poderá sugerir a ausência de sintomas de discalculia.

Através de uma análise destes dois gráficos é possível verificar que o Adolfo concluiu as suas tarefas com tempo inferiores à média da turma em todas as áreas de dificuldade em que foi testado.

Figure 56 - Student results analysis screen preview

This page presents the user with the comparison of the student's results with the results achieved by his class. This page is divided into two sections, one that compares scores and one that compares time taken to complete certain tasks.

These sections can be seen in the following figures (Figure 57 and Figure 58):



Através de uma análise visual destes dois gráficos é possível verificar que o Adolfo tem pontuções superiores à média da turma nas área de contagem, contas com vírgulas e outros.
É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pelo Adolfo se encontram dentro da média da turma nas área de contas algébricas e unidades/medidas.
Estes resultados indicam que o Adolfo não tem dificuldades quando comparado com os seus colegas de turma, um facto que poderá sugerir a ausência de sintomas de discalculia.

Figure 57 - Student results analysis close-up 1

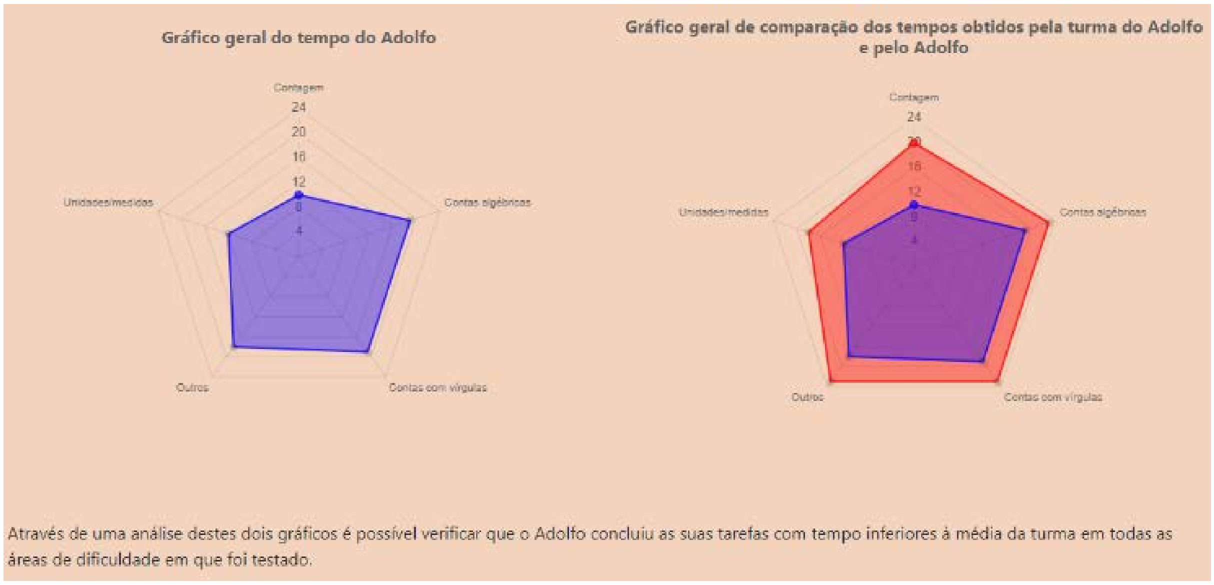


Gráfico geral do tempo do Adolfo
Gráfico geral de comparação dos tempos obtidos pela turma do Adolfo e pelo Adolfo
Contagem
Contas algébricas
Contas com vírgulas
Outros
Unidades/medidas
24
20
16
12
8
4

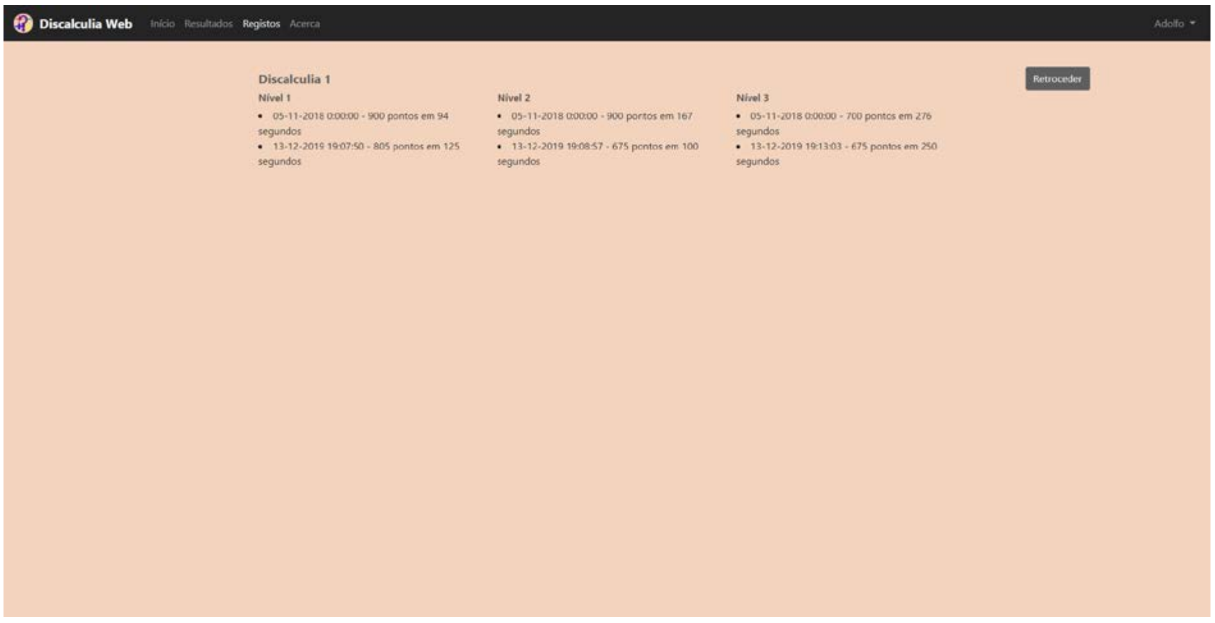
Através de uma análise destes dois gráficos é possível verificar que o Adolfo concluiu as suas tarefas com tempo inferiores à média da turma em todas as áreas de dificuldade em que foi testado.

Figure 58 - Student results analysis close-up 2

The graphs for each section show the mean score of the student of his class, as well as the time that was taken, in different tasks related to a certain area of dyscalculia. On the left, the student can see his graphs, and on the left the user can see the student's graph compared to his class's.

The text section describes the student's results opposed to those of the student's class. It can be seen

as the text version of the graphs, where the areas of dyscalculia where the student achieved better, worse or normal results compared to his class, are explicitly written and explained.
The last paragraph may tell the student, or not, depending on the results that were achieved, to further investigate each individual result in order to understand the nature of the difficulties that were detected when compared to his class.
On the right side of the initial page (Figure 51), there is the Ver mais registos button, that allows the user to reach all of the student's scores. They can also be reached by pressing the Registos option on the top left side of the screen (Figure 52). The page that is reached is the following (Figure 59):



Discalculia 1

Nivel 1
- 05-11-2018 0:00:00 - 900 pontos em 94 segundos
- 13-12-2019 19:07:50 - 805 pontos em 125 segundos

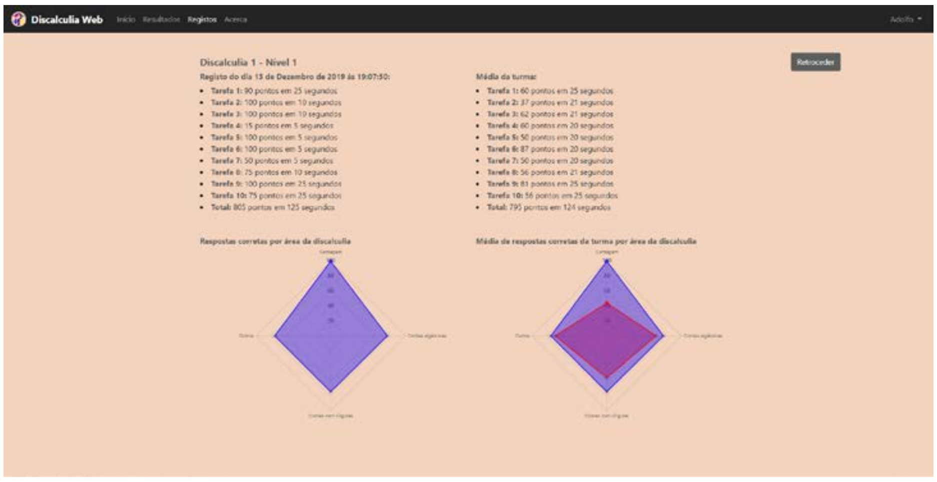
Nivel 2
- 05-11-2018 0:00:00 - 900 pontos em 167 segundos
- 13-12-2019 19:08:57 - 675 pontos em 100 segundos

Nivel 3
- 05-11-2018 0:00:00 - 700 pontos em 276 segundos
- 13-12-2019 19:13:03 - 675 pontos em 250 segundos

Figure 59 - Student scores screen preview

This page divides the students' scores into games, in the example (Figure 59) there is only one, and each game into levels, represented by each column. These columns include all scores achieved in a game's level and shows its date, as well as the time and score achieved in each attempt. There also is a shadow that appears when the user hovers the score with the mouse point in order to indicate that the score is clickable.

If the user clicks on a score in the previous page (Figure 59) or on the right side of the initial page (Figure 51), the following page is shown (Figure 60, Figure 61 and Figure 62):



Discalculia 1 - Nível 1

Registo do dia 13 de Dezembro de 2019 às 19:07:50:

- **Tarefa 1:** 90 pontos em 25 segundos
- **Tarefa 2:** 100 pontos em 10 segundos
- **Tarefa 3:** 100 pontos em 10 segundos
- **Tarefa 4:** 15 pontos em 5 segundos
- **Tarefa 5:** 100 pontos em 5 segundos
- **Tarefa 6:** 100 pontos em 5 segundos
- **Tarefa 7:** 50 pontos em 5 segundos
- **Tarefa 8:** 75 pontos em 10 segundos
- **Tarefa 9:** 100 pontos em 25 segundos
- **Tarefa 10:** 75 pontos em 25 segundos
- **Total:** 805 pontos em 125 segundos

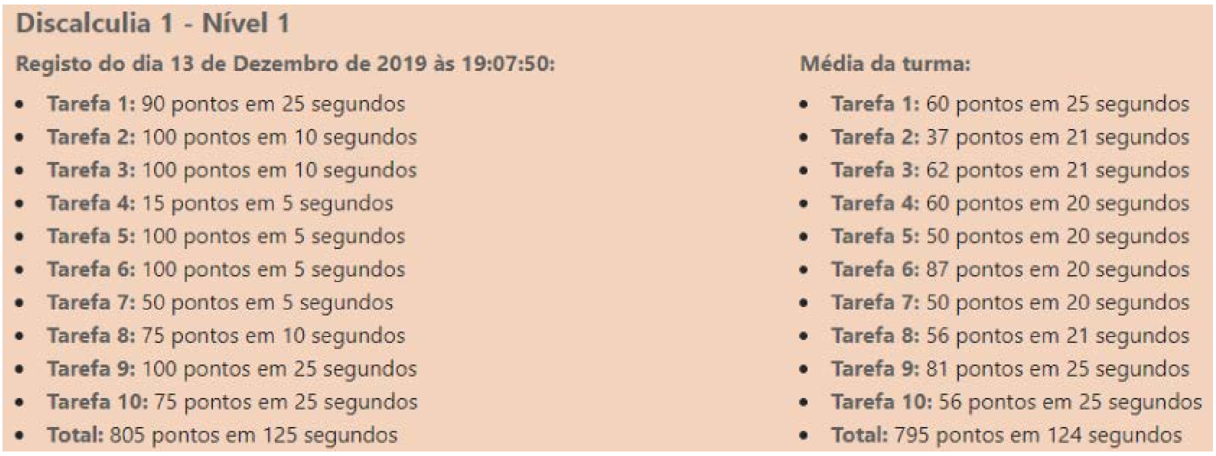
Média da turma:

- **Tarefa 1:** 60 pontos em 25 segundos
- **Tarefa 2:** 37 pontos em 21 segundos
- **Tarefa 3:** 62 pontos em 21 segundos
- **Tarefa 4:** 60 pontos em 20 segundos
- **Tarefa 5:** 50 pontos em 20 segundos
- **Tarefa 6:** 87 pontos em 20 segundos
- **Tarefa 7:** 50 pontos em 20 segundos
- **Tarefa 8:** 56 pontos em 21 segundos
- **Tarefa 9:** 81 pontos em 25 segundos
- **Tarefa 10:** 56 pontos em 25 segundos
- **Total:** 795 pontos em 124 segundos

Respostas corretas por área da discalculia

Média de respostas corretas da turma por área da discalculia

Figure 60 - Student score analysis screen preview



Discalculia 1 - Nível 1

Registo do dia 13 de Dezembro de 2019 às 19:07:50:

- **Tarefa 1:** 90 pontos em 25 segundos
- **Tarefa 2:** 100 pontos em 10 segundos
- **Tarefa 3:** 100 pontos em 10 segundos
- **Tarefa 4:** 15 pontos em 5 segundos
- **Tarefa 5:** 100 pontos em 5 segundos
- **Tarefa 6:** 100 pontos em 5 segundos
- **Tarefa 7:** 50 pontos em 5 segundos
- **Tarefa 8:** 75 pontos em 10 segundos
- **Tarefa 9:** 100 pontos em 25 segundos
- **Tarefa 10:** 75 pontos em 25 segundos
- **Total:** 805 pontos em 125 segundos

Média da turma:

- **Tarefa 1:** 60 pontos em 25 segundos
- **Tarefa 2:** 37 pontos em 21 segundos
- **Tarefa 3:** 62 pontos em 21 segundos
- **Tarefa 4:** 60 pontos em 20 segundos
- **Tarefa 5:** 50 pontos em 20 segundos
- **Tarefa 6:** 87 pontos em 20 segundos
- **Tarefa 7:** 50 pontos em 20 segundos
- **Tarefa 8:** 56 pontos em 21 segundos
- **Tarefa 9:** 81 pontos em 25 segundos
- **Tarefa 10:** 56 pontos em 25 segundos
- **Total:** 795 pontos em 124 segundos

Figure 61 - Student score analysis 1

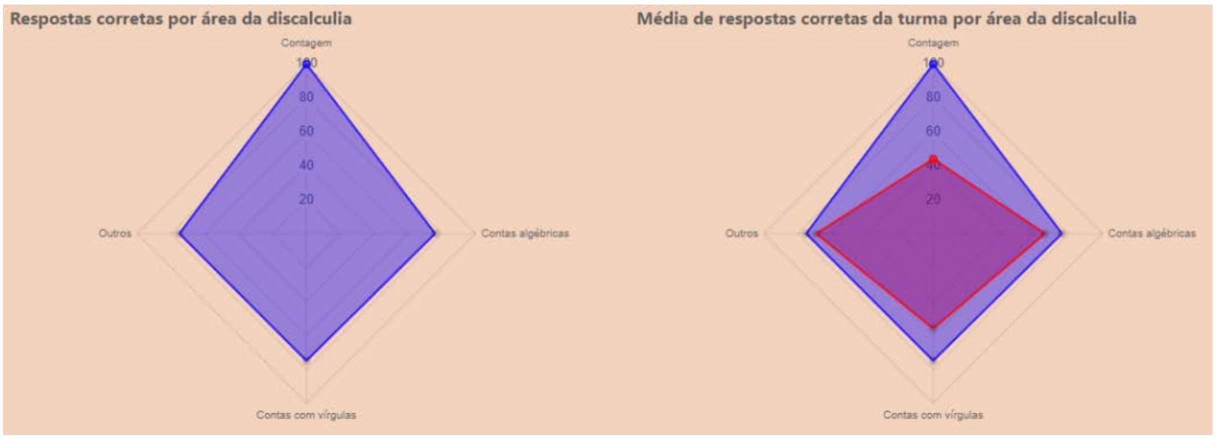


Figure 62 - Student score analysis 2

This page is divided into two sections. The first section (Figure 61) shows the scores and times that the student achieved in each task of the selected level score record on the left side, and the scores and times mean of the student's class's records, on the left side.
The second section (Figure 62) shows two graphs, similar to those presented in the student results analysis (Figure 56), but directed towards the selected level. The left graph shows the scores achieved by the student in the different areas of dyscalculia that the selected level tests, and the right graph compares the student's results to those of the student's class's mean.
Moving on to the teacher's side of the web application, after a successful login as a teacher by selecting the correct option (Figure 40), the user is presented the following screen (Figure 63):

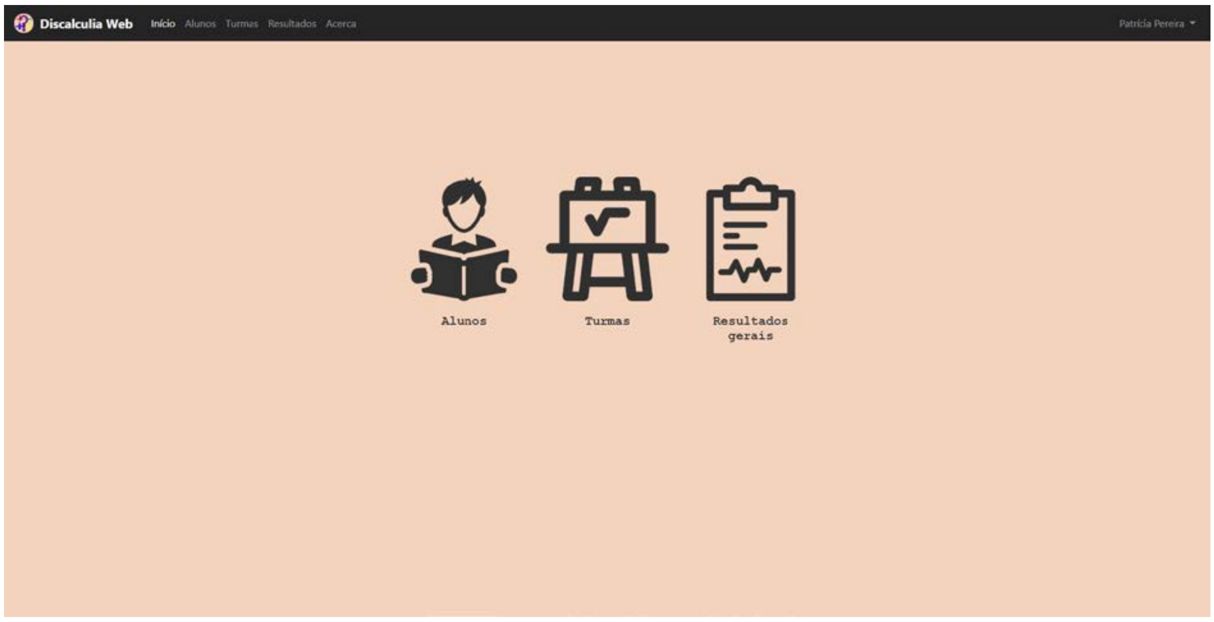


Figure 63 – Teacher screen preview

On the top left of the screen the user is presented the web application brand as well as six different options (Figure 64):

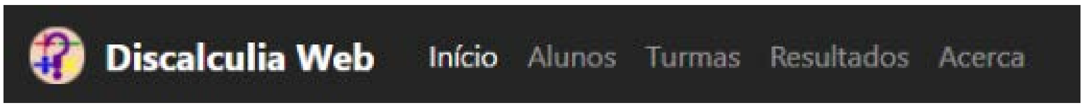


Figure 64 - Teacher screen top left close-up

These options lead the user to different pages according to the option that was selected. *Início* leads the user to the initial teacher page (Figure 63), *Alunos* leads to the page where the teacher can see all of his students. *Turmas* allows the user to see and access all of the teacher's classes. *Resultados* leads to the page where the teacher can see an analysis of all of his classes' results. *Acerca* makes the pop-up that explains the project appear on screen (Figure 45).

On the top right side of the screen the user is presented the name of the teacher that he is logged in as and is shown two further options if he clicks on the name, just like the student's case (Figure 53), as can be seen in the following figure (Figure 65):

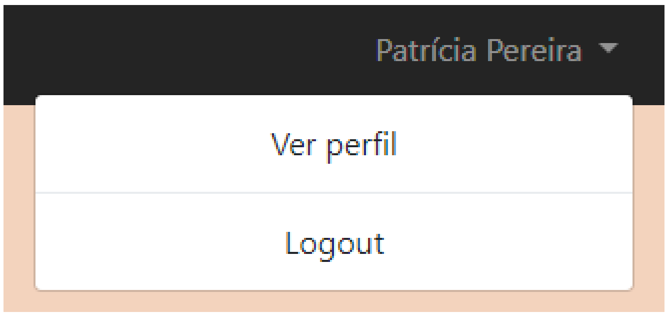
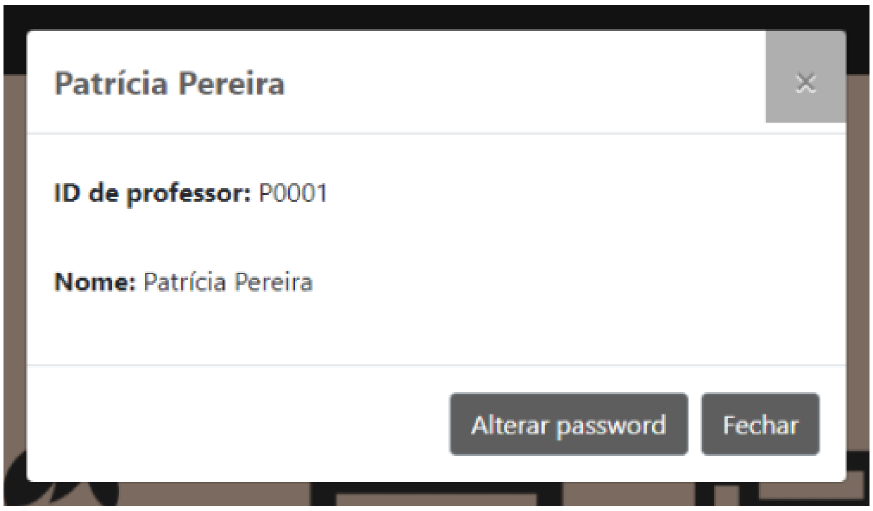


Figure 65 - Teacher screen top right side close-up

The second option, Logout, allow the user to logout of the teacher's account and return to the initial screen (Figure 40). The first option, *Ver perfil*, allows the user to the see the teacher's profile and the information that is stored regarding the teacher's identity (Figure 66):


Patrícia Pereira

ID de professor: P0001

Nome: Patrícia Pereira

Figure 66 - Teacher profile close-up

Here, the user can see the teacher's identification code and name. On the bottom of the pop-up there are two options. *Fechar*, that allows the user to close the pop-up, and *Alterar password* that allows the user to change the teacher's password similarly to the student's case (Figure 55), as can be seen in the following figure (Figure 67):

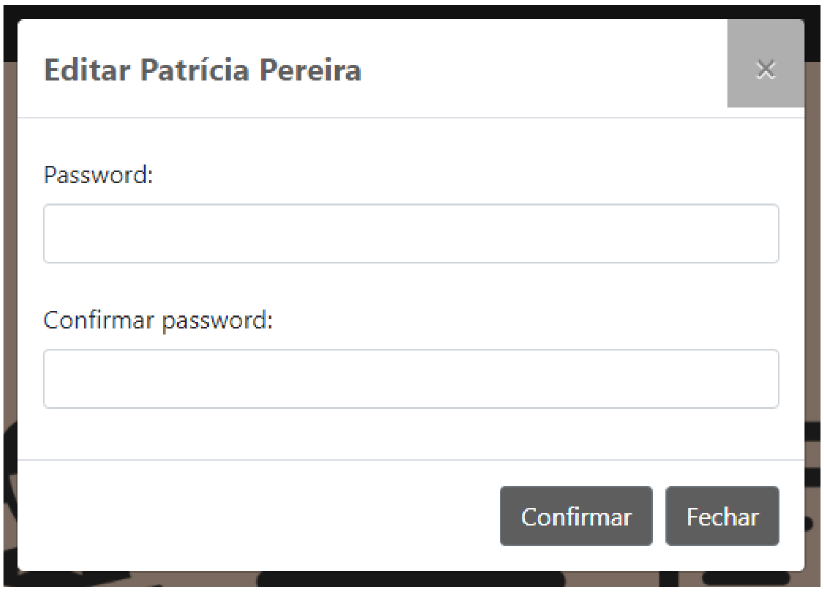


Figure 67 - Teacher password edition pop-up close-up

By clicking in the *Alunos* option, present on the initial screen (Figure 63) or on the top left of the screen (Figure 64), the user is shown the following page (Figure 68):

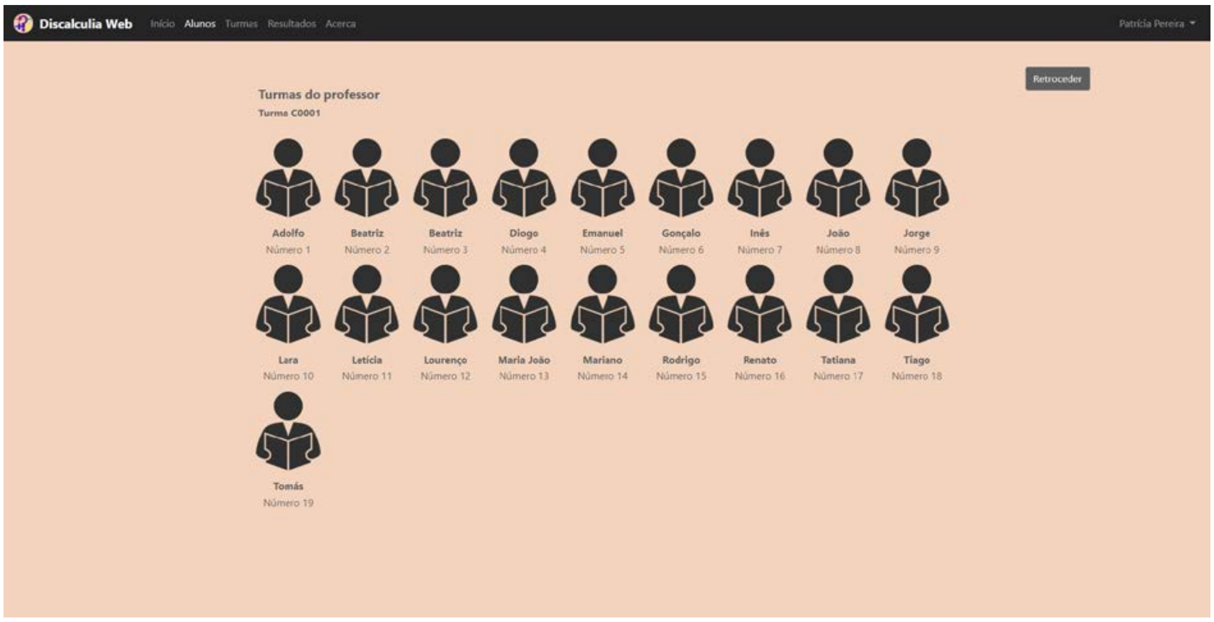
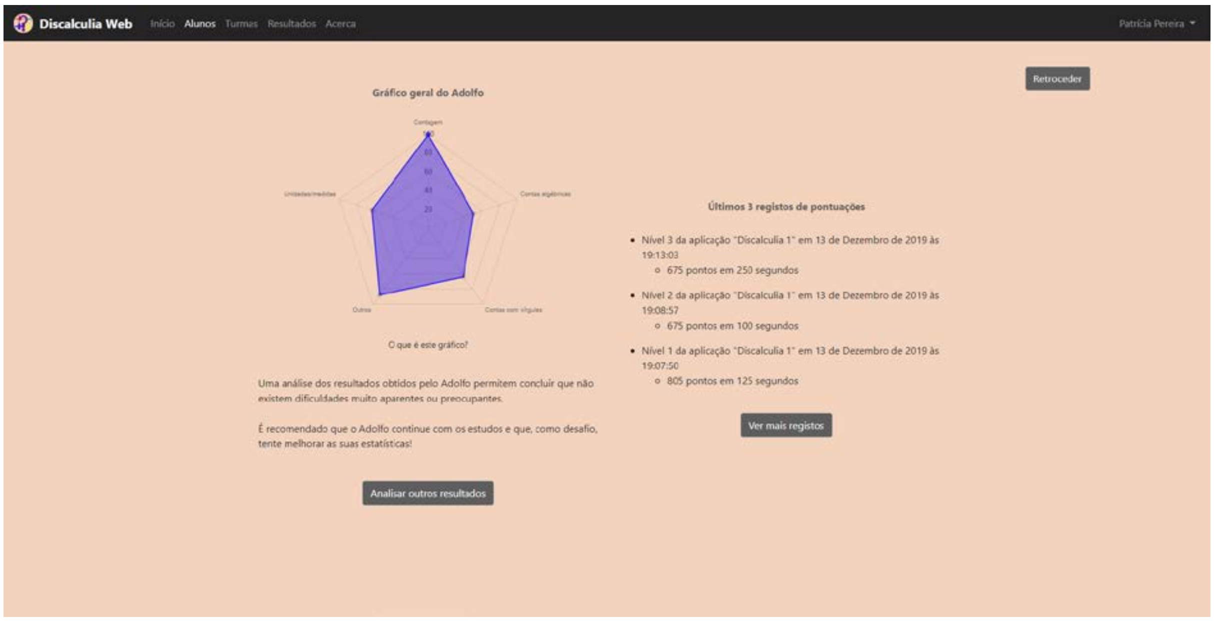


Figure 68 - Teacher students screen preview

In this page, the user can see all of his students divided into their respective class. An important note that must be made since it is one of the main differences between the teacher and the expert user, is the fact that the teacher can only see students from his classes while the expert can see every existing student. In this case, the teacher can see each student from his classes, identified by his name and class number. If the user hovers over a student with the mouse pointer, the student's image is pushed forward, and a shadow is added in order to indicate the fact that each student is clickable.

After clicking a student, the teacher can access all of the student's screens, starting with the initial screens (Figure 48 and Figure 49), as can be seen in the following figure (Figure 69):



Uma análise dos resultados obtidos pelo Adolfo permitem concluir que não existem dificuldades muito aparentes ou preocupantes.

É recomendado que o Adolfo continue com os estudos e que, como desafio, tente melhorar as suas estatísticas!

Últimos 3 registos de pontuações

- Nível 3 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:13:03
  - 675 pontos em 250 segundos
- Nível 2 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:08:57
  - 675 pontos em 100 segundos
- Nível 1 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:07:50
  - 805 pontos em 125 segundos

Figure 69 - Teacher student screen preview

In order to access the following screen (Figure 70), the user can click the *Turmas* option on the initial screen (Figure 63) or on the top left of the screen (Figure 64):

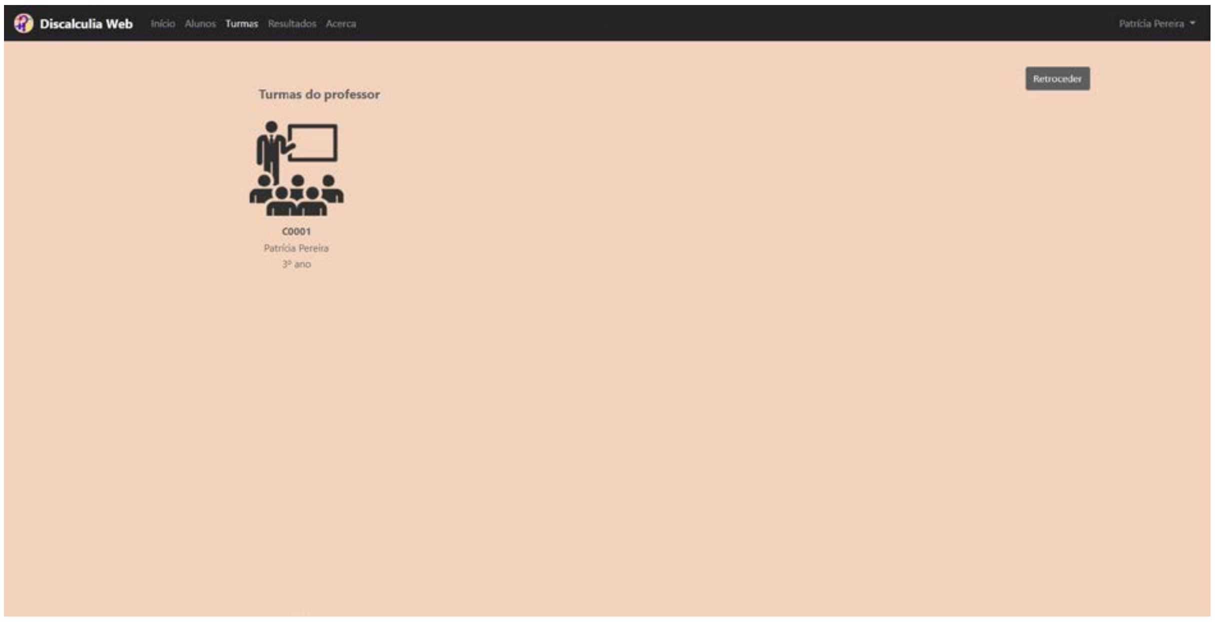
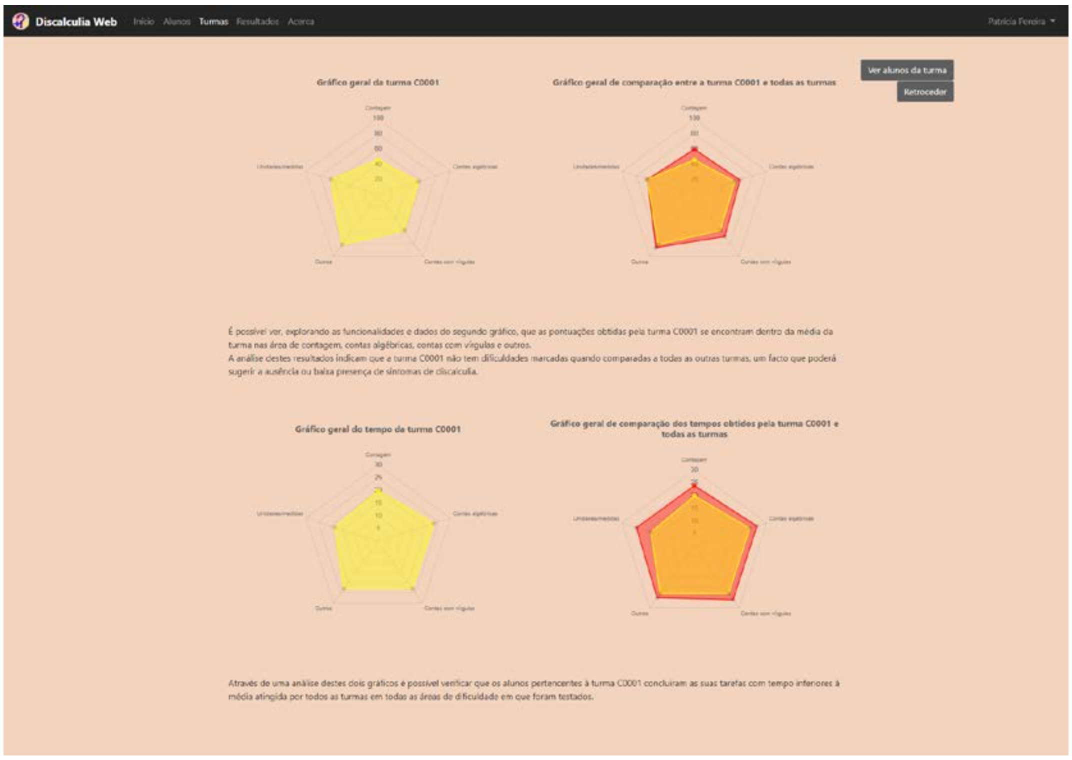


Figure 70 - Teacher classes screen preview

In this screen the user can see all of the teacher's classes, not all classes, identified by their identification code, the teacher's name and the scholarship level. By hovering with the mouse over the class, the image is pushed forward, and a shadow is added in order to show the user that the class is clickable.

By clicking a class, the user can see an analysis of the class's performance, similar to that of the student's (Figure 56), as can be seen in the following figures (Figure 71, Figure 72 and Figure 73):



Discalculia Web
Início
Alunos
Turmas
Resultados
Acerca
Patrícia Ferreira
Ver alunos da turma
Retroceder
Gráfico geral da turma C0001
Gráfico geral de comparação entre a turma C0001 e todas as turmas
Contagem
Contas algébricas
Contas com vírgulas
Outros
Unidades/medidas
Gráfico geral do tempo da turma C0001
Gráfico geral de comparação dos tempos obtidos pela turma C0001 e todas as turmas

É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pela turma C0001 se encontram dentro da média da turma nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que a turma C0001 não tem dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes à turma C0001 concluiram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 71 - Teacher class screen preview

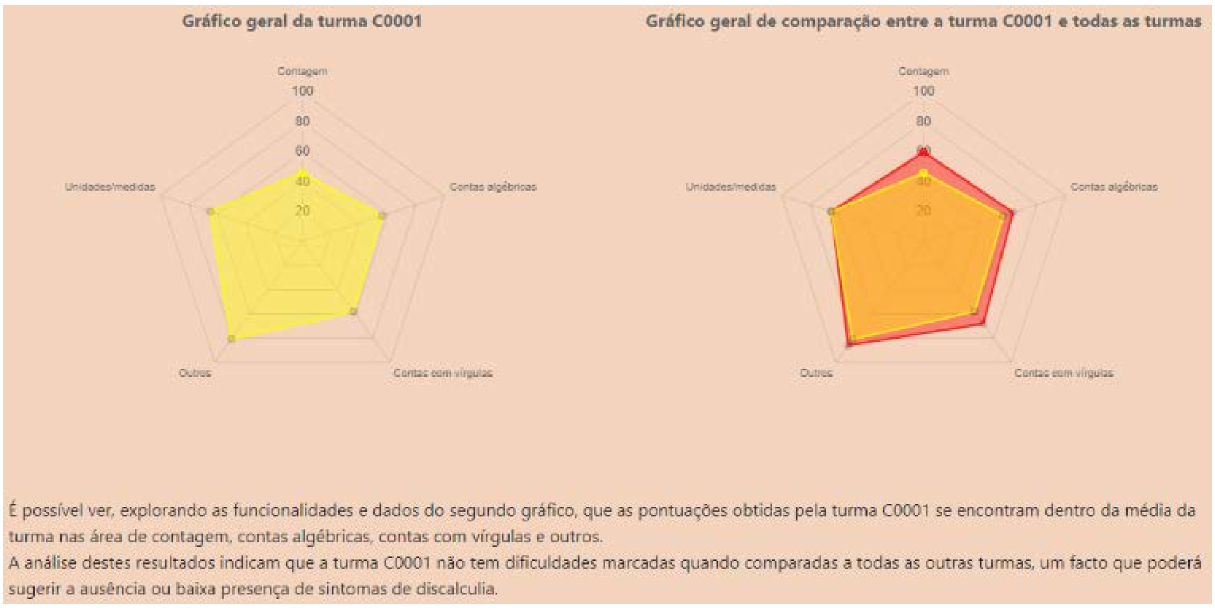
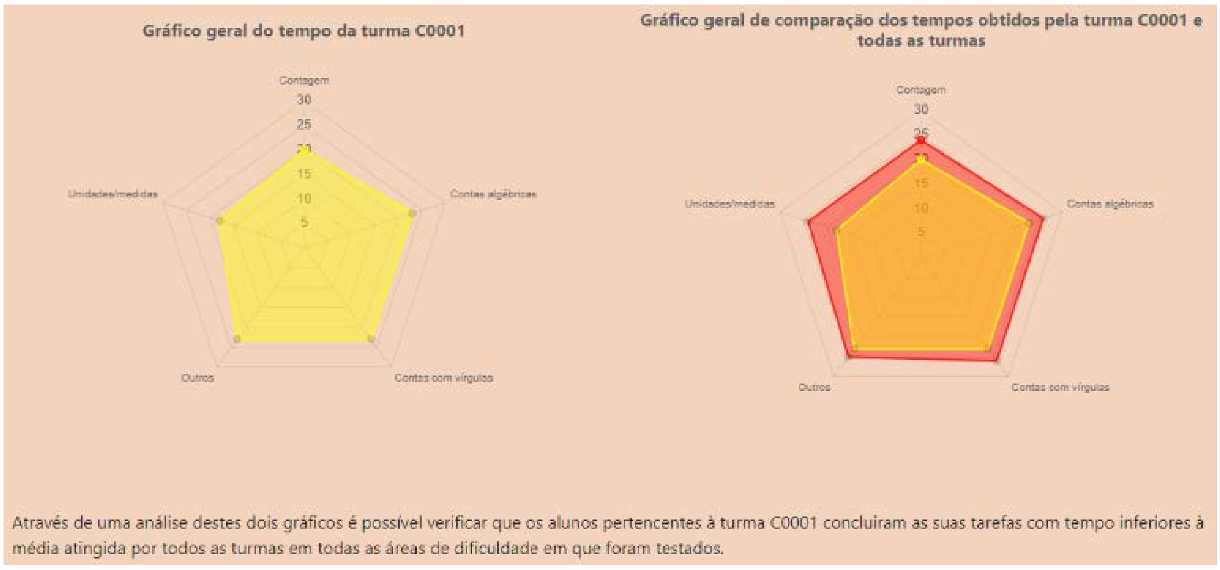


Gráfico geral da turma C0001
Gráfico geral de comparação entre a turma C0001 e todas as turmas
Contagem
100
80
60
40
20
Contas algébricas
Contas com vírgulas
Outros
Unidades/medidas

É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pela turma C0001 se encontram dentro da média da turma nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que a turma C0001 não tem dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

Figure 72 - Teacher class 1



Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes à turma C0001 concluiram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 73 - Teacher class 2

This page, just like the student results analysis page (Figure 56), is divided into two sections. The first section (Figure 72), shows the score mean achieved by the selected class in the different tasks regarding the various areas of dyscalculia and compares it to the score mean achieved by all the students. There is also a written analysis of these scores that explains the areas where the selected class was better, worse or within the score mean of all students.

The second section (Figure 73) shows two graphs that display the time mean achieved by the selected class in the different areas of dyscalculia and also compares it to the time mean achieved by all existing students, accompanied by a written analysis, similar to the one described in the previous paragraph.

By selecting the *Ver alunos da turma* button present on the top right of the selected class's screen (Figure 71), the user is directed to the following page (Figure 74):

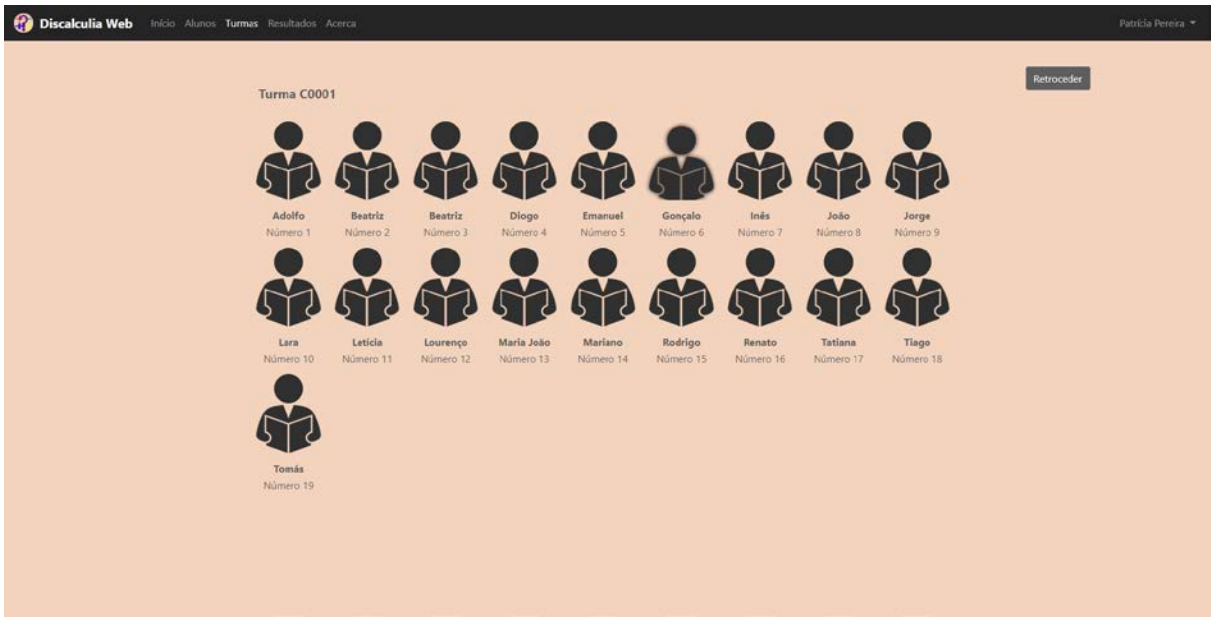
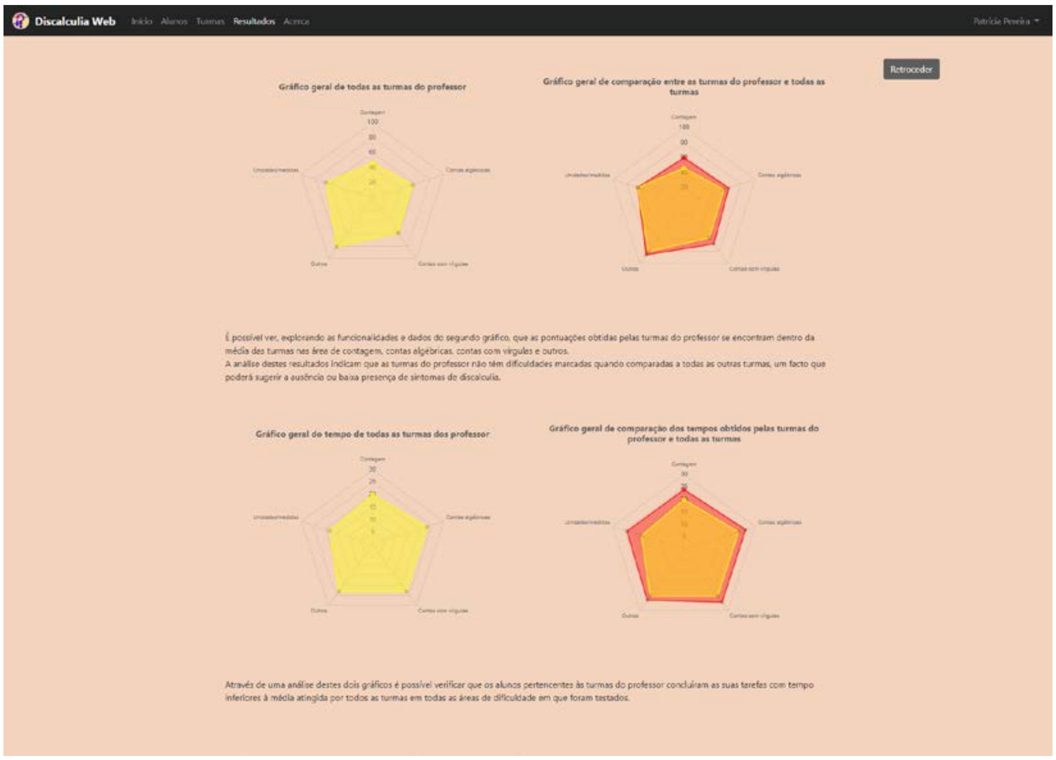


Figure 74 - Teacher class students screen preview

This screen is very similar to the screen that contains all of the students that belong to the teacher's classes (Figure 68) with the only difference being the number of classes that are displayed. While the first contains all of the teacher's classes, this one only shows the students of one of the teacher's classes. Each student is also clickable and leads to the selected student's screens (Figure 69).

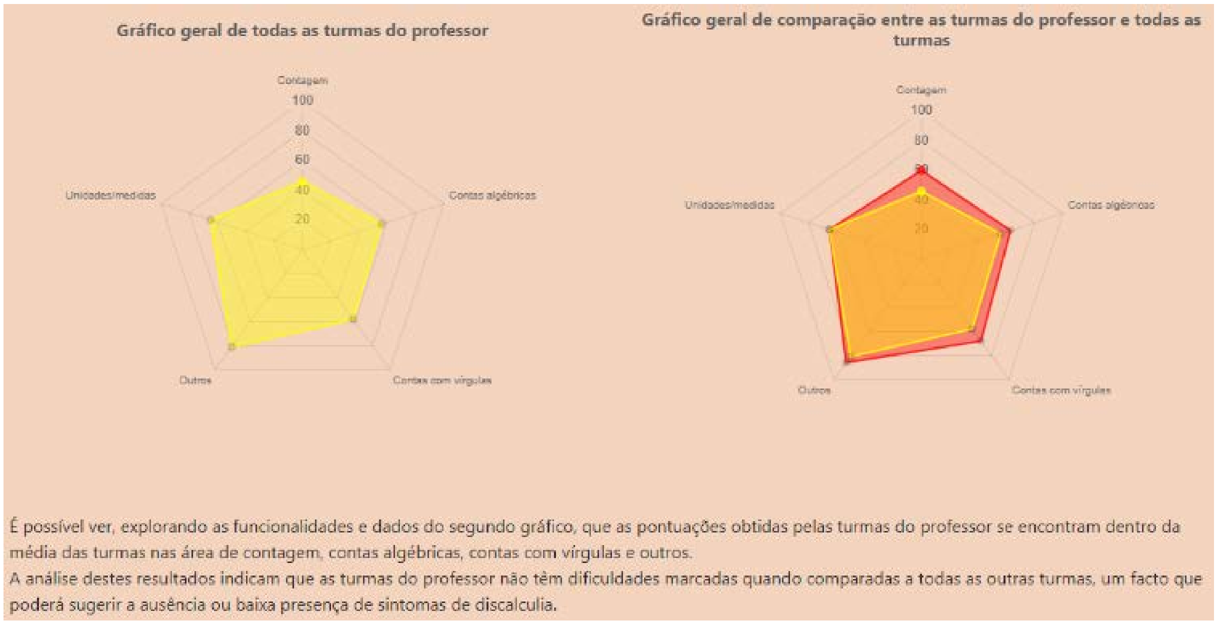
The last of the teacher's screens (Figure 75, Figure 76 and Figure 77), can be reached by selecting the *Resultados* option on the top right of the screen (Figure 64) or by accessing the Resultados gerais option on the initial screen of the teacher user type (Figure 60):


Discalculia Web
Início
Alunos
Turmas
Resultados
Acerca
Patrícia Pereira
Retroceder
Gráfico geral de todas as turmas do professor
Gráfico geral de comparação entre as turmas do professor e todas as turmas
Gráfico geral do tempo de todas as turmas dos professor
Gráfico geral de comparação dos tempos obtidos pelas turmas do professor e todas as turmas
Contagem
Contas algébricas
Contas com vírgulas
Outros
Unidades/medidas

É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pelas turmas do professor se encontram dentro da média das turmas nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que as turmas do professor não têm dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

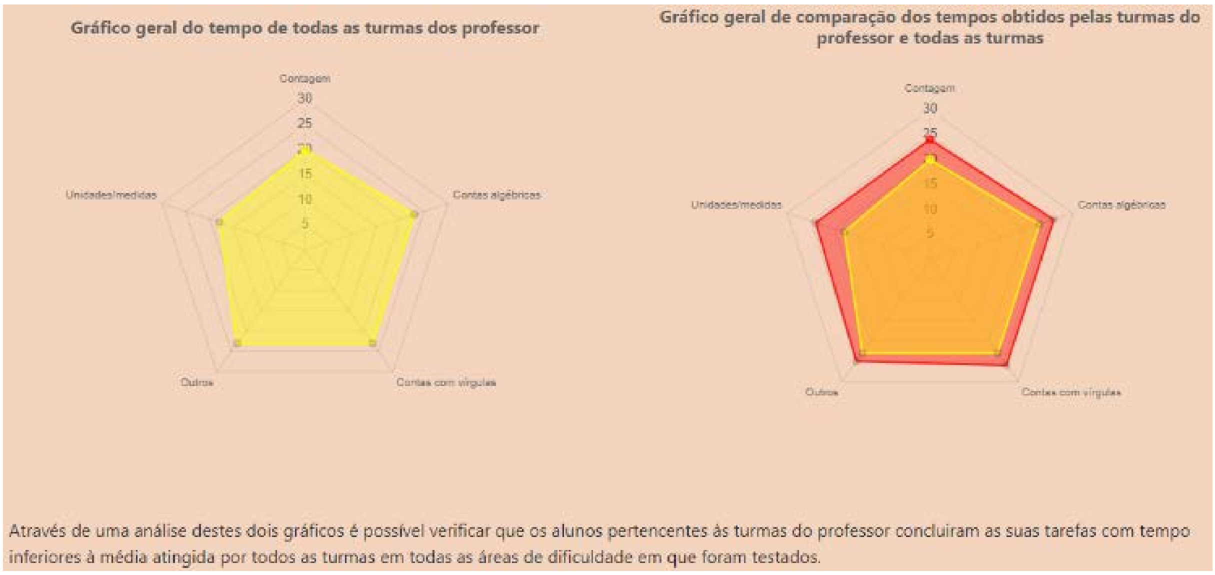
Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes às turmas do professor concluíram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 75 - Teacher classes analysis screen preview



É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pelas turmas do professor se encontram dentro da média das turmas nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que as turmas do professor não têm dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

Figure 76 - Teacher classes analysis 1



Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes ás turmas do professor concluiram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 77 - Teacher classes analysis 2

Thanks to these screens, the teacher can analyze all of his classes' performances and compare it to those of all the students. The first set of graphs (Figure 76), compare the teacher's classes' mean scores on different areas of dyscalculia to the score means achieved by all the students, while the second set of graphs (Figure 77) does the same but with time means. These sets of graphs are accompanied by a written analysis that compare these scores and times by saying in which areas the teacher's classes are better, worse and within the mean of all the students.

By logging in to the web application as an expert, the user is presented the following screen (Figure 78):

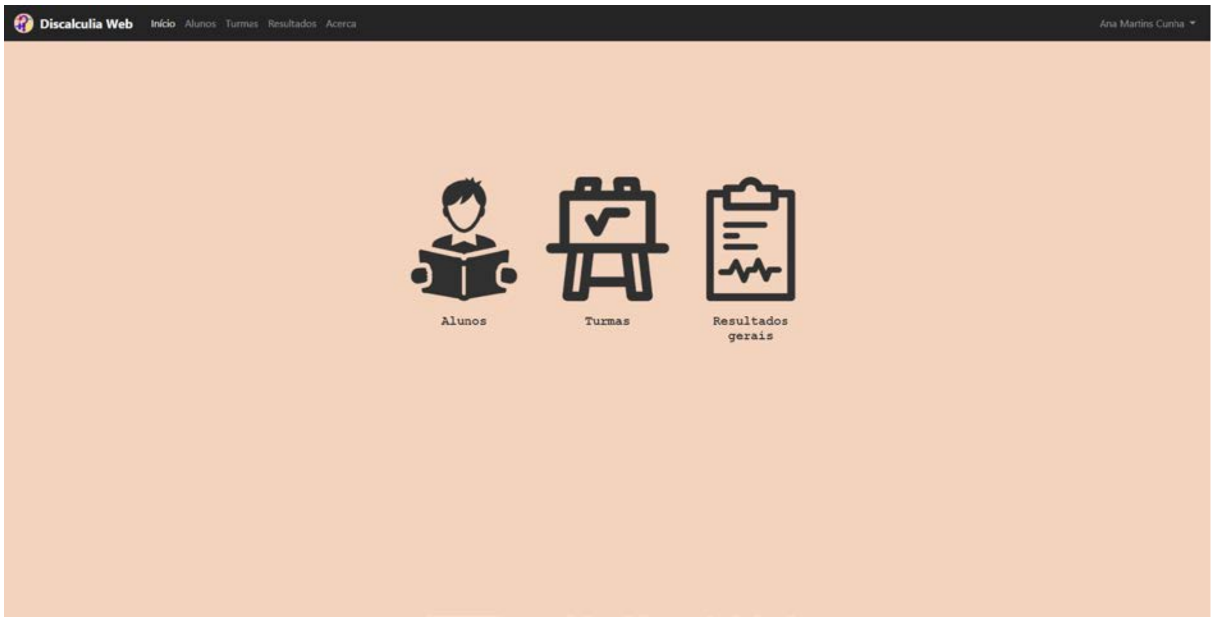


Figure 78 - Expert screen preview

These options are exactly like the options present of the teacher's initial screen (Figure 63) as well as the options present on the top right of the screen (Figure 79):

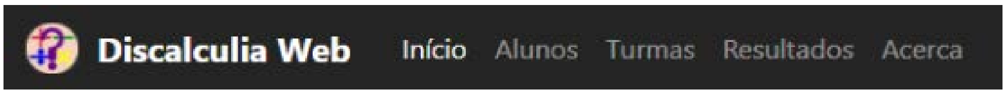


Figure 79 - Expert screen top left close-up

These options lead the user to different pages according to the option that was selected. *Início* leads the user to the initial expert page (Figure 78), *Alunos* leads to the page where the expert can see all students. *Turmas* allows the user to see and access all classes. *Resultados* and *Resultados gerais* leads to the page where the expert can see an analysis of all of his classes' results. *Acerca* makes the pop-up that explains the project appear on screen (Figure 45).

Just like the teacher's case (Figure 65), on the top right of the screen the expert can see his profile by selecting the Ver perfil option or logout to the web application's initial screen (Figure 40), as can be seen in the following close-up (Figure 80):

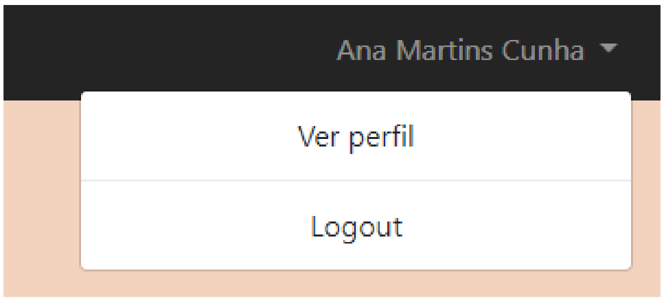
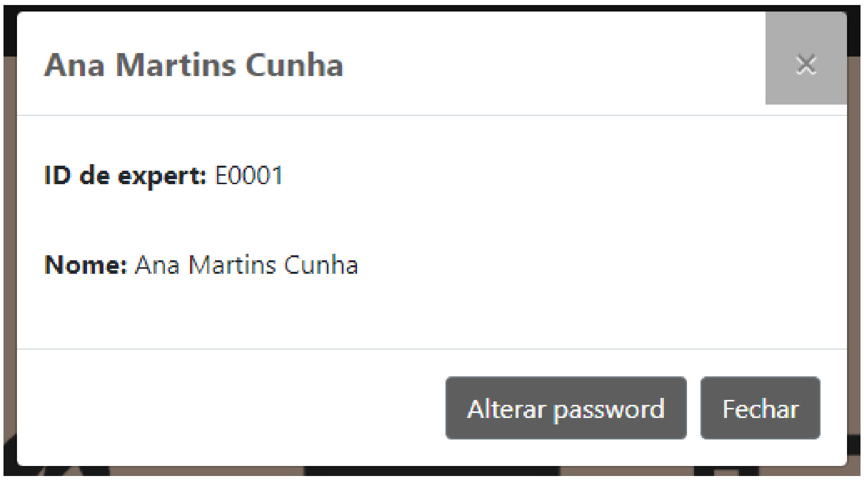


Figure 80 - Expert screen top right close-up

By selecting the *Ver perfil* option the expert can see his profile, like follows (Figure 81):


Ana Martins Cunha

ID de expert: E0001

Nome: Ana Martins Cunha

Figure 81 - Expert profile close-up

Here, the user can see the expert's identification code, as well as the expert's name. Also, as can be seen, there are two buttons, the *Fechar* button, that allows the user to close the pop-up, and the *Alterar password* button which allows the expert to change login password just like the student's case (Figure 55).

By clicking in the *Alunos* option present on the initial screen (Figure 78) or on the top left of the screen (Figure 79), the user is shown the following page (Figure 82):

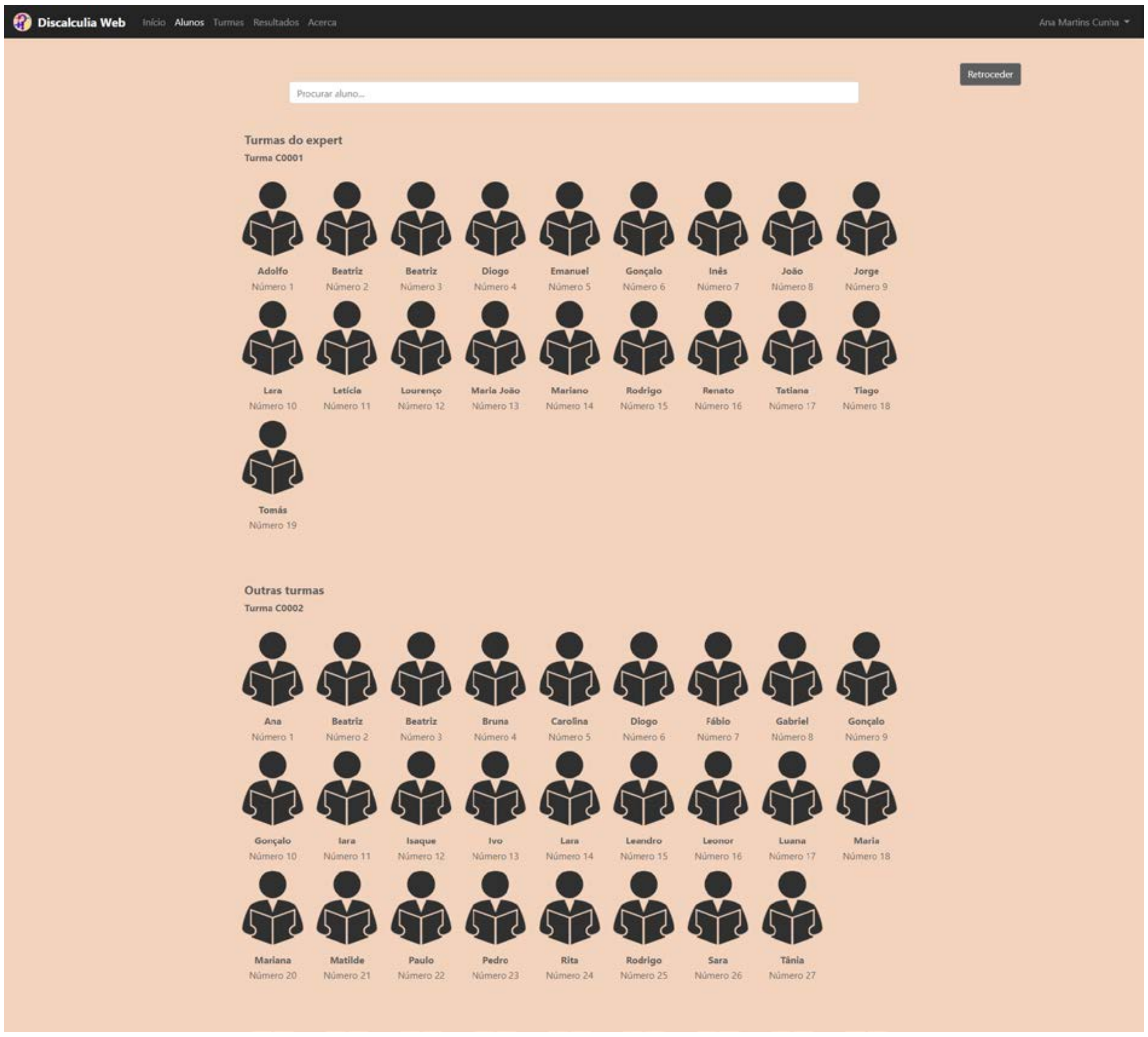
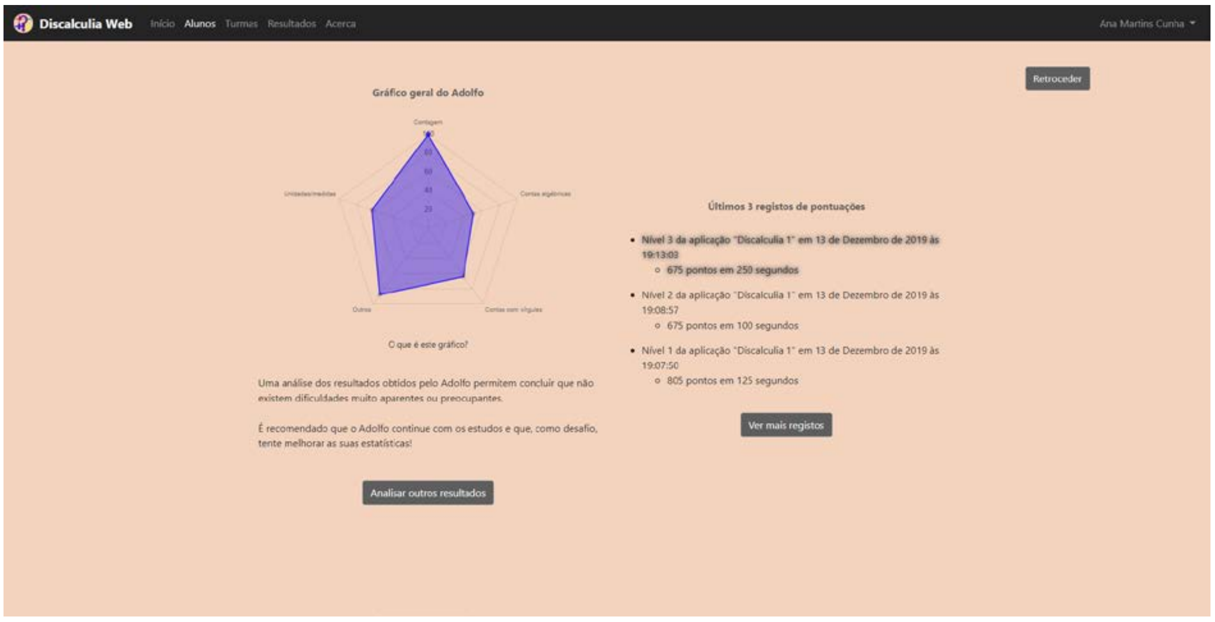


Figure 82 - Expert students screen preview

In this page, all existing students are presented to the user divided into classes that are, in turn, divided into a logged in expert's classes section and a section containing all other classes.

Each student is clickable, an action that is shown to be possible by pushing forward the student's image and adding a shadow to it when hovering the mouse over it.

By clicking the student's image, the expert has access to all of the selected student's screen, as seen in the following screen (Figure 83):



Uma análise dos resultados obtidos pelo Adolfo permitem concluir que não existem dificuldades muito aparentes ou preocupantes.

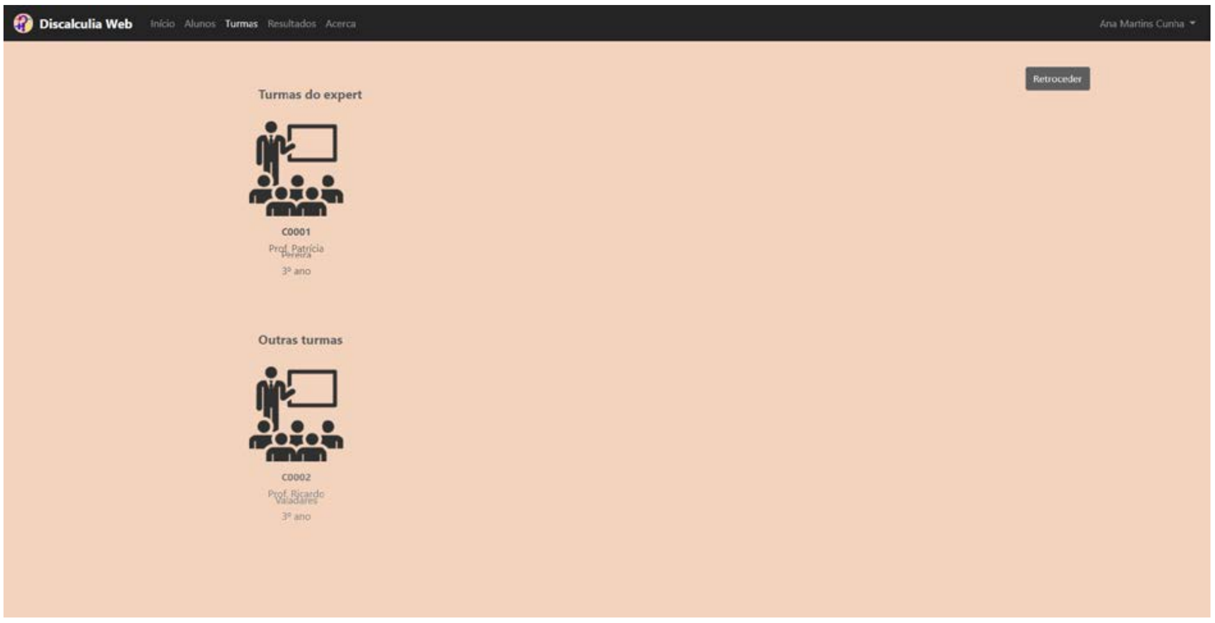
É recomendado que o Adolfo continue com os estudos e que, como desafio, tente melhorar as suas estatísticas!

Últimos 3 registos de pontuações

- Nível 3 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:13:03
  - 675 pontos em 250 segundos
- Nível 2 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:08:57
  - 675 pontos em 100 segundos
- Nível 1 da aplicação "Discalculia 1" em 13 de Dezembro de 2019 às 19:07:50
  - 805 pontos em 125 segundos

Figure 83 - Expert student screen preview

By selecting the *Turmas* option on top of the screen (Figure 79) or on the initial expert screen (Figure 78), the user has access to the following screen (Figure 84):



Turmas do expert

C0001
Prof. Patrícia Pereira
3º ano

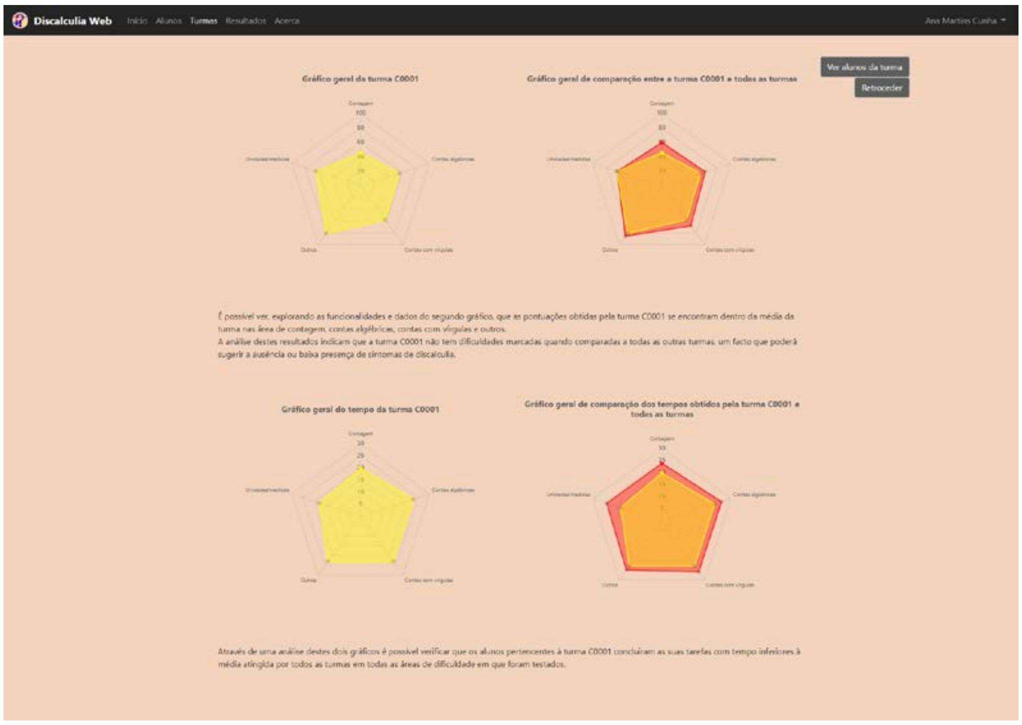
Outras turmas

C0002
Prof. Ricardo Valadares
3º ano

Figure 84 - Expert classes screen preview

In this screen the user has access to all existing classes divided into two sections, the expert's classes and all other classes. Every class is clickable, a possibility indicated, again, by the addition of a shadow to the class image and by pushing the image forward.

That action leads to the following page (Figure 85, Figure 86 and Figure 87):

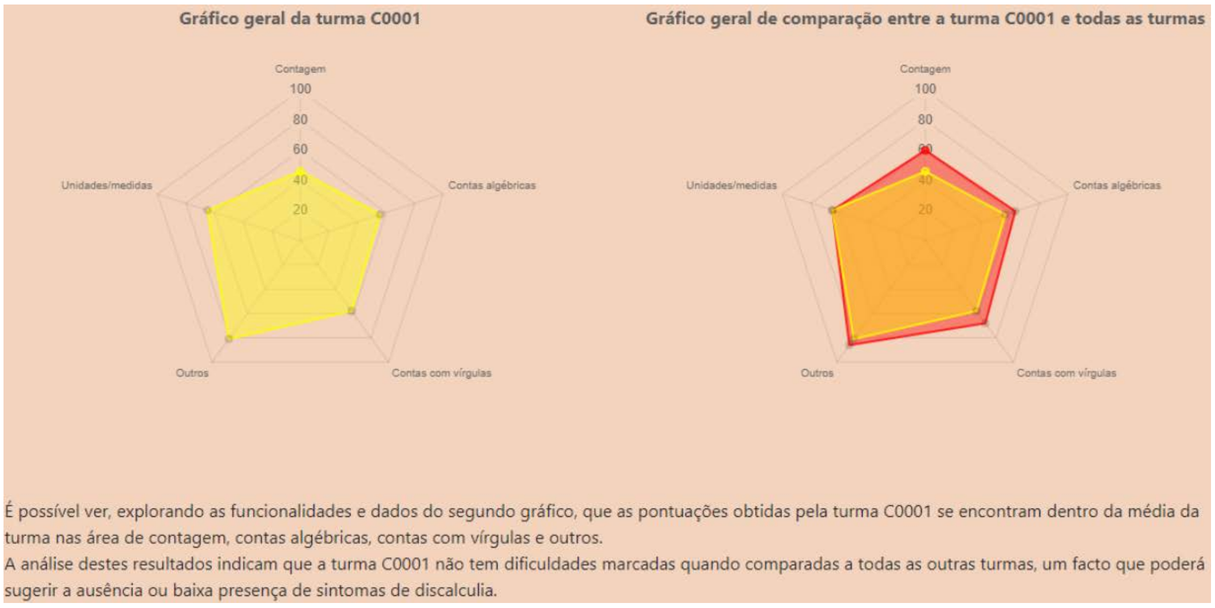


Discalculia Web
Início
Alunos
Turmas
Resultados
Acerca
Ana Martins Cunha
Ver alunos da turma
Retroceder
Gráfico geral da turma C0001
Gráfico geral de comparação entre a turma C0001 e todas as turmas
Gráfico geral do tempo da turma C0001
Gráfico geral de comparação dos tempos obtidos pela turma C0001 e todas as turmas

É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pela turma C0001 se encontram dentro da média da turma nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que a turma C0001 não tem dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

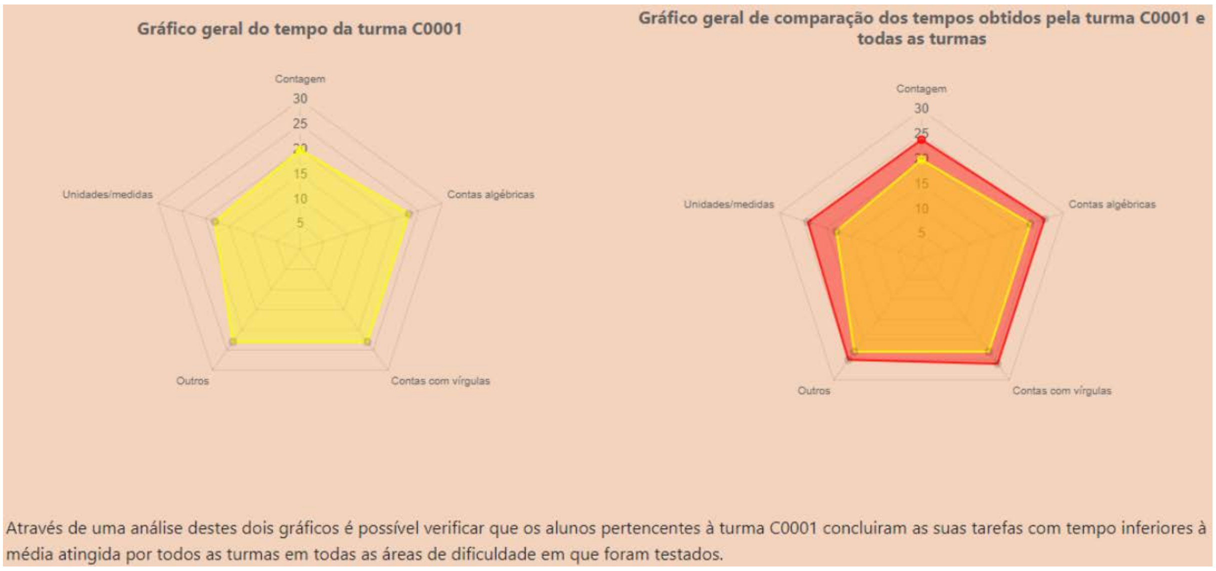
Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes à turma C0001 concluíram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 85 - Expert class screen preview



É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pela turma C0001 se encontram dentro da média da turma nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que a turma C0001 não tem dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

Figure 86 - Expert class 1



Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes à turma C0001 concluiram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 87 - Expert class 2

This page analyzes the selected class's performance compared to the one achieved by all students. It is divided into two sections.

The first section (Figure 86) presents the user with two graphs. The first graph shows the selected class's mean score in the existing areas of dyscalculia, while the second graph compares it to all students' mean score in said dyscalculia areas. These graphs are also accompanied by a written analysis that explains the areas where the selected class is better, worse or within the score mean of all students.

The second section (Figure 87) also presents the user with similar graphs and written analysis similar to the previous one but directed towards the mean time taken to solve tasks directed towards a certain area of dyscalculia.

By selecting the *Ver alunos da turma* button that exists on the top right of the previous screen (Figure 85), the user is presented the following screen (Figure 88):

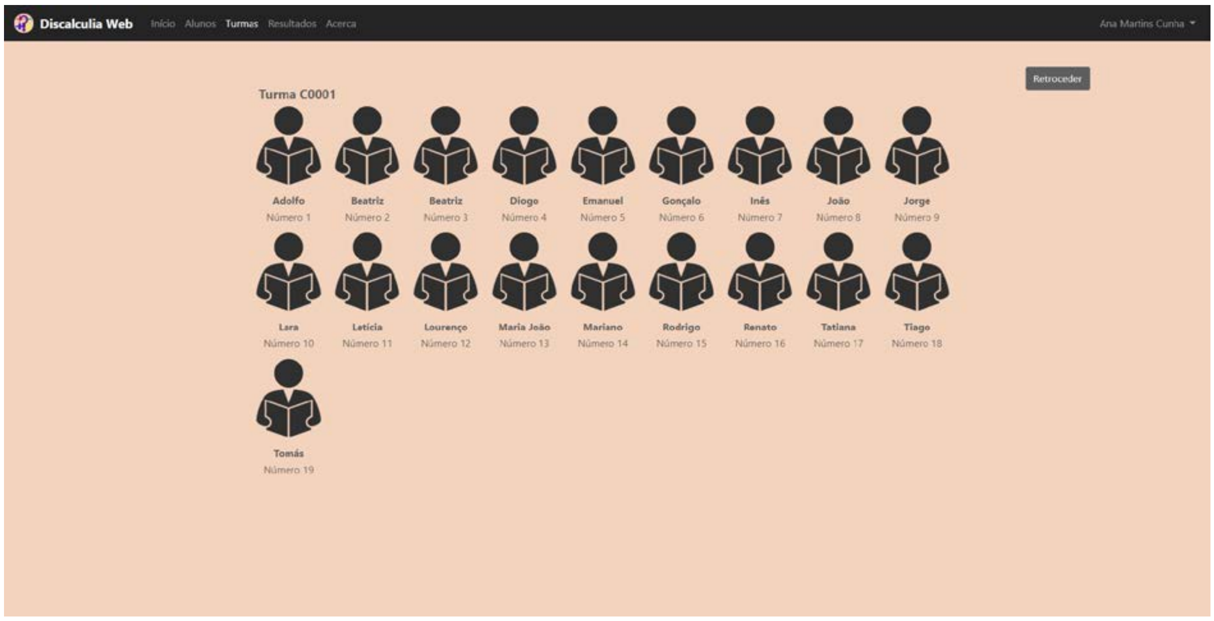
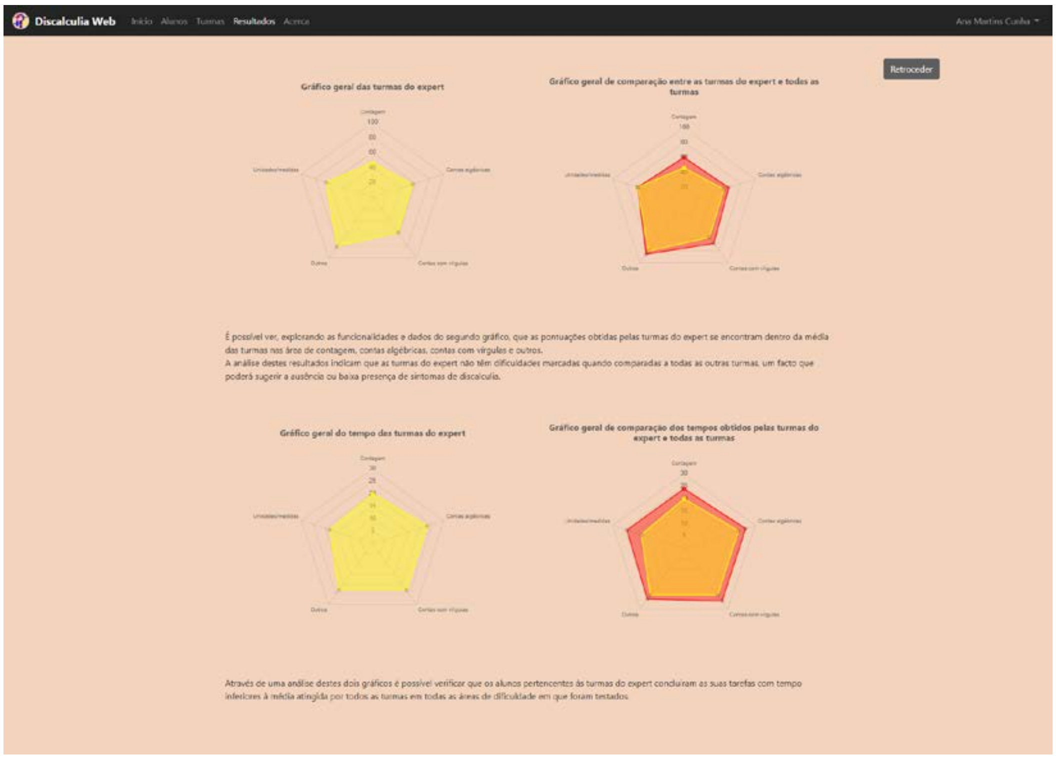


Figure 88 - Expert class students screen preview

This screen allows the user to see all the students from the select class and, by clicking on their respective image, access the view that student has, just like with the expert student screen (Figure 83).

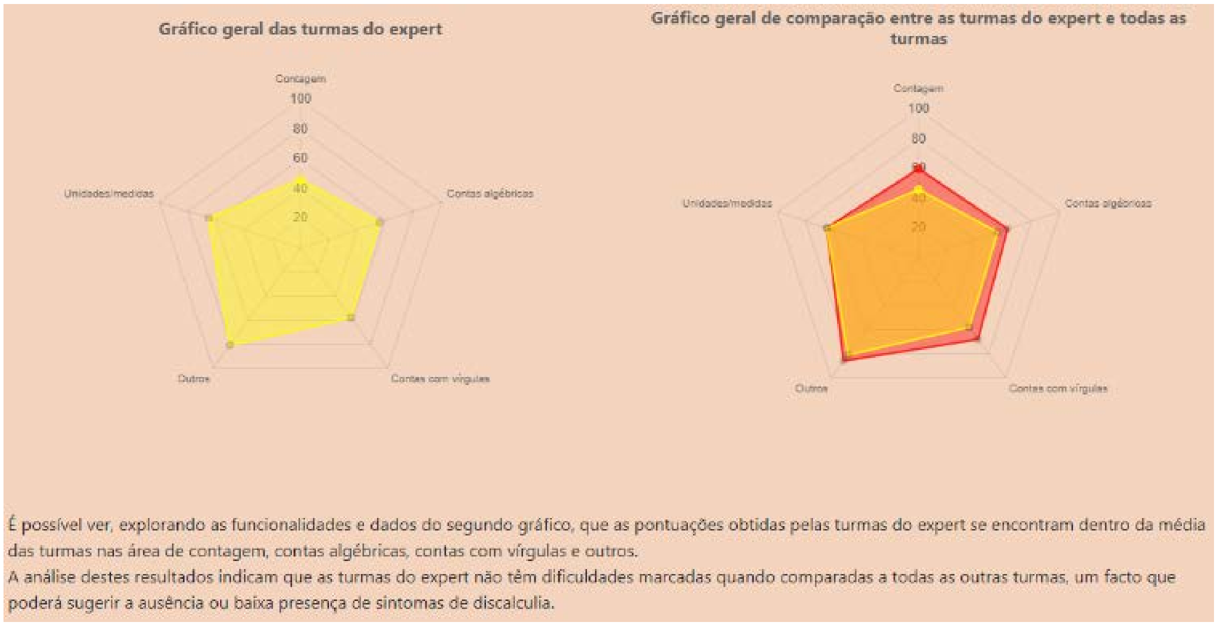
The final screen the expert user has access to, accessible by clicking the *Resultados* option on the top left of the screen (Figure 79) or the *Resultados gerais* option on the initial expert screen (Figure 78), is as follows (Figure 89, Figure 90 and Figure 91):


Discalculia Web
Início
Alunos
Turmas
Resultados
Acerca
Ana Martins Cunha
Retroceder
Gráfico geral das turmas do expert
Gráfico geral de comparação entre as turmas do expert e todas as turmas
Contagem
Contas algébricas
Contas com vírgulas
Outros
Unidades/medidas
Gráfico geral do tempo das turmas do expert
Gráfico geral de comparação dos tempos obtidos pelas turmas do expert e todas as turmas

É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pelas turmas do expert se encontram dentro da média das turmas nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que as turmas do expert não têm dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

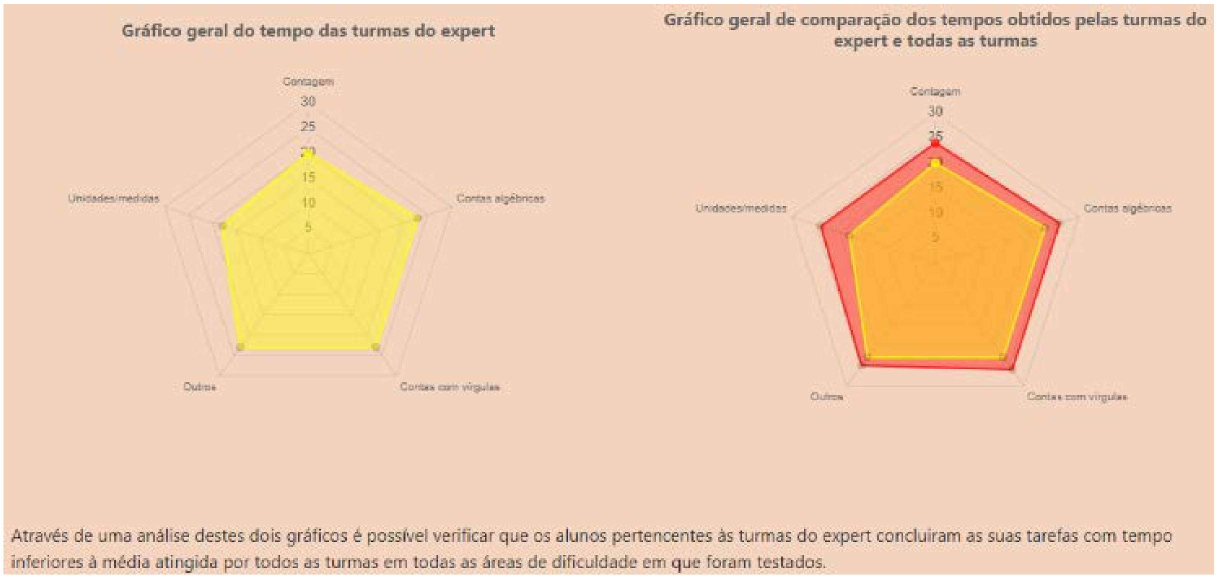
Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes às turmas do expert concluíram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 89 - Expert classes analysis screen preview



É possível ver, explorando as funcionalidades e dados do segundo gráfico, que as pontuações obtidas pelas turmas do expert se encontram dentro da média das turmas nas área de contagem, contas algébricas, contas com vírgulas e outros.
A análise destes resultados indicam que as turmas do expert não têm dificuldades marcadas quando comparadas a todas as outras turmas, um facto que poderá sugerir a ausência ou baixa presença de sintomas de discalculia.

Figure 90 - Expert classes analysis 1



Através de uma análise destes dois gráficos é possível verificar que os alunos pertencentes às turmas do expert concluiram as suas tarefas com tempo inferiores à média atingida por todos as turmas em todas as áreas de dificuldade em que foram testados.

Figure 91 - Expert classes analysis 2

These screens allow the expert to analyze all of his classes' performance and compare it to all the performance of all the students. The first two graphs (Figure 90), show the score mean of all the expert's classes on different areas of dyscalculia and compares it to score mean of all students. They are accompanied by a written comparison of those graphs. The last two graphs (Figure 91), show the time mean of all of the expert's classes and compares them to the mean time of all the students. Again, a written analysis of the comparison these last set of graphs make accompanies them.

For the last user type, the administrator, selected in the initial screen the web application presents the user (Figure 40), the user is shown the following screen (Figure 92):

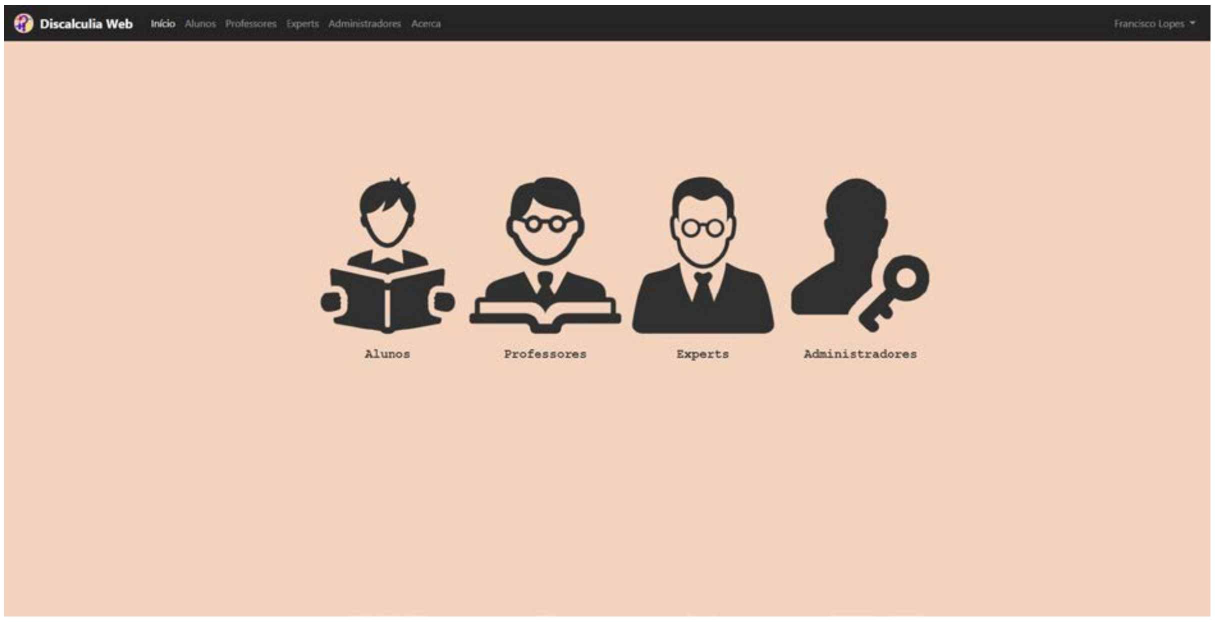


Figure 92 - Administrator screen preview

On the top left of the screen the user is presented the same options as the previous screen, like follows (Figure 93):

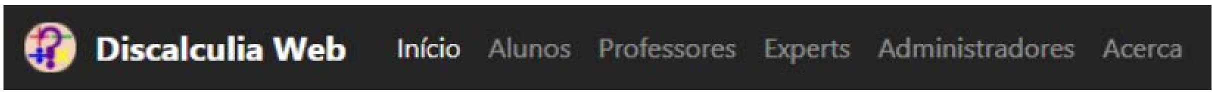


Figure 93 - Administrator screen top left close-up

The *Início* option leads the user to the initial screen (Figure 92), the *Alunos* option leads to a page containing all students, the *Professores* option leads to a page that contains all teachers, the *Experts* option leads the user to the page that shows all experts, the *Administradores* option leads to the page containing all administrators and the *Acerca* option makes the pop-up that explains the project appear on screen (Figure 45).

On the top right side of the screen the user is presented the name of the administrator that he is logged in as and is shown two further options if he clicks on the name, just like with all the other user types, as can be seen in the following figure (Figure 94):

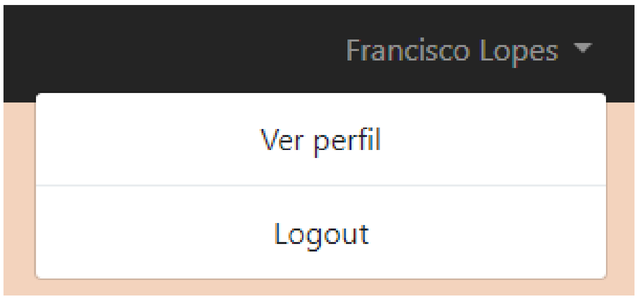
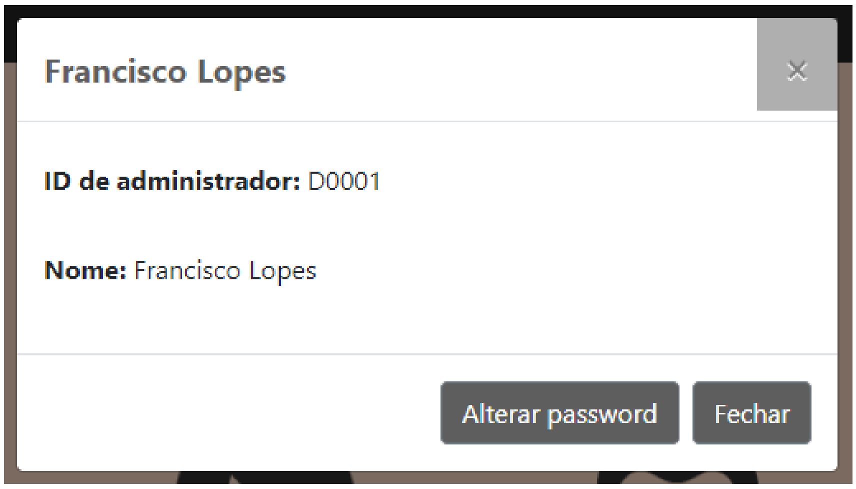


Figure 94 - Administrator screen top right side close-up

The second option, Logout, allow the user to logout of the teacher's account and return to the initial screen (Figure 40). The first option, *Ver perfil*, allows the user to the see the logged in administrator's profile and the information that is stored regarding his identity (Figure 95):



ID de administrador: D0001

Nome: Francisco Lopes

Figure 95 - Administrator profile close-up

Here, the user can see the administrator's identification code and name. On the bottom of the pop-up there are two options. *Fechar*, that allows the user to close the pop-up, and *Alterar password*.

Alterar password allows the user to change the administrator's password similarly to the other user types, as can be seen in the next figure (Figure 96).

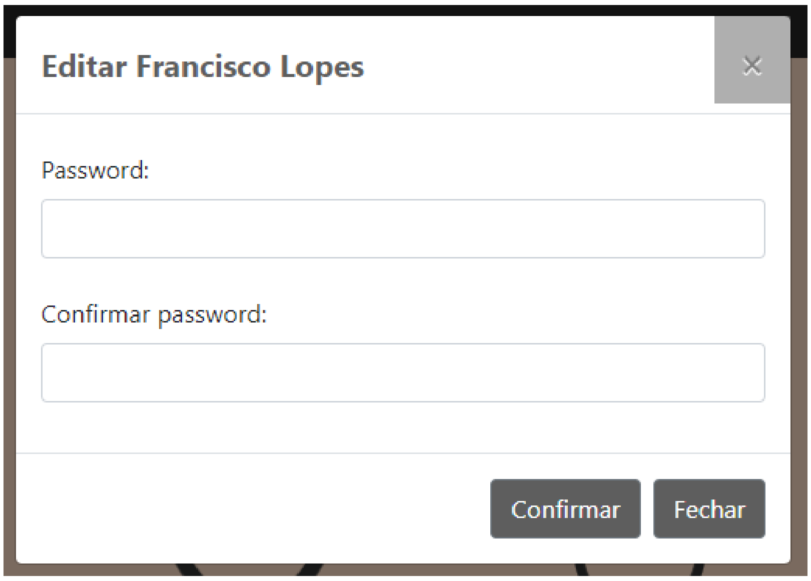


Figure 96 - Administrator password edition pop-up close-up

By clicking the *Alunos* option, present on the initial administrator screen (Figure 92) or on the top left of the screen (Figure 93), the administrator is given access to the following page (Figure 97):

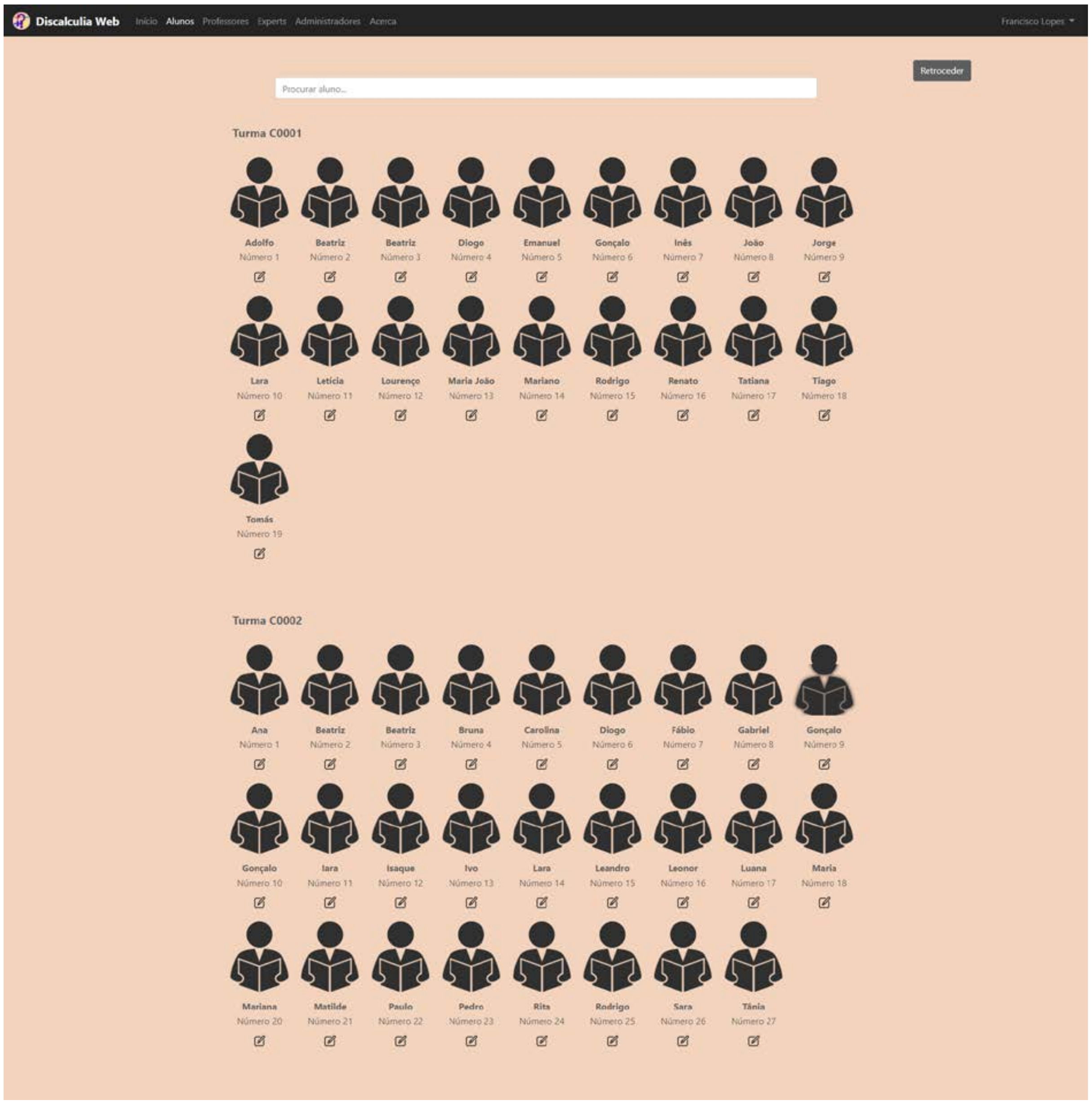
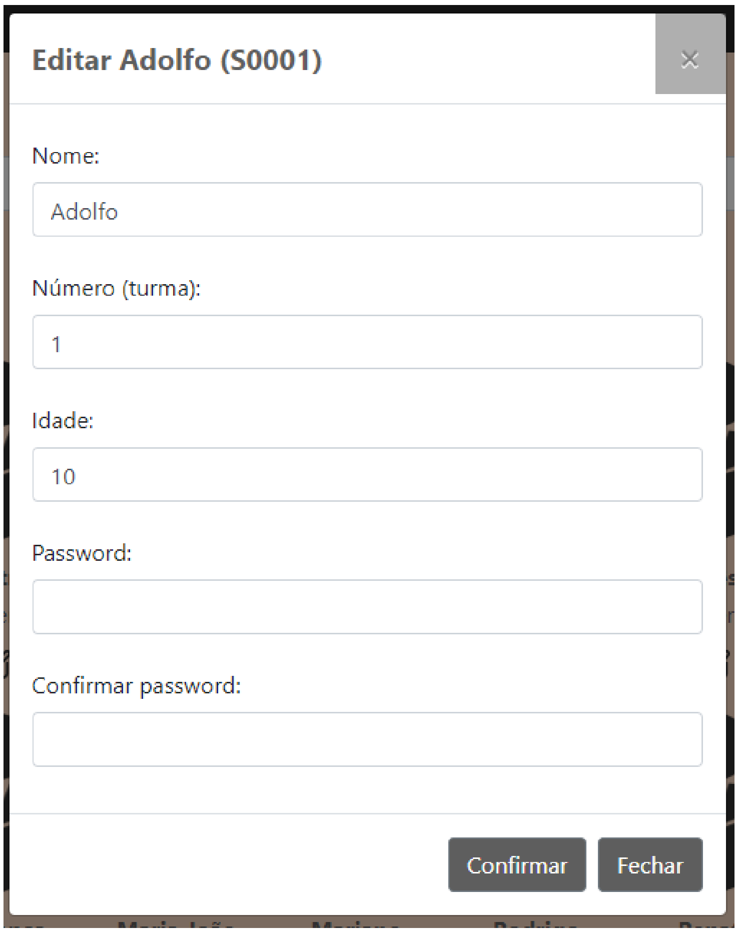


Figure 97 - Administrator students screen preview

In this screen, the administrator has access to all students, as well as their view, by clicking on the student's image, just like with the teacher and expert's case, but with the ability to edit the student's details.

By clicking on the image that indicates the editing action, present below the student's details, there is a pop up that appears that allow the user to change the student's detail (Figure 98), as can be seen as follows:


Editar Adolfo (S0001)

Nome:
Adolfo

Número (turma):
1

Idade:
10

Password:

Confirmar password:

Figure 98 - Administrator student edition close-up

This pop-up loads the existing student data, the student's name, class number and age, and presents it to the user, with the possibility to edit those details. There are also two fields that allow the administrator to change the student's password, by typing it in the first box and repeating it in the second, an action that is useful in the case the student's user forgets the password.

By clicking the *Professores* option, present on the initial administrator screen (Figure 92) or on the top left of the screen (Figure 93), the administrator is shown the following page (Figure 99):

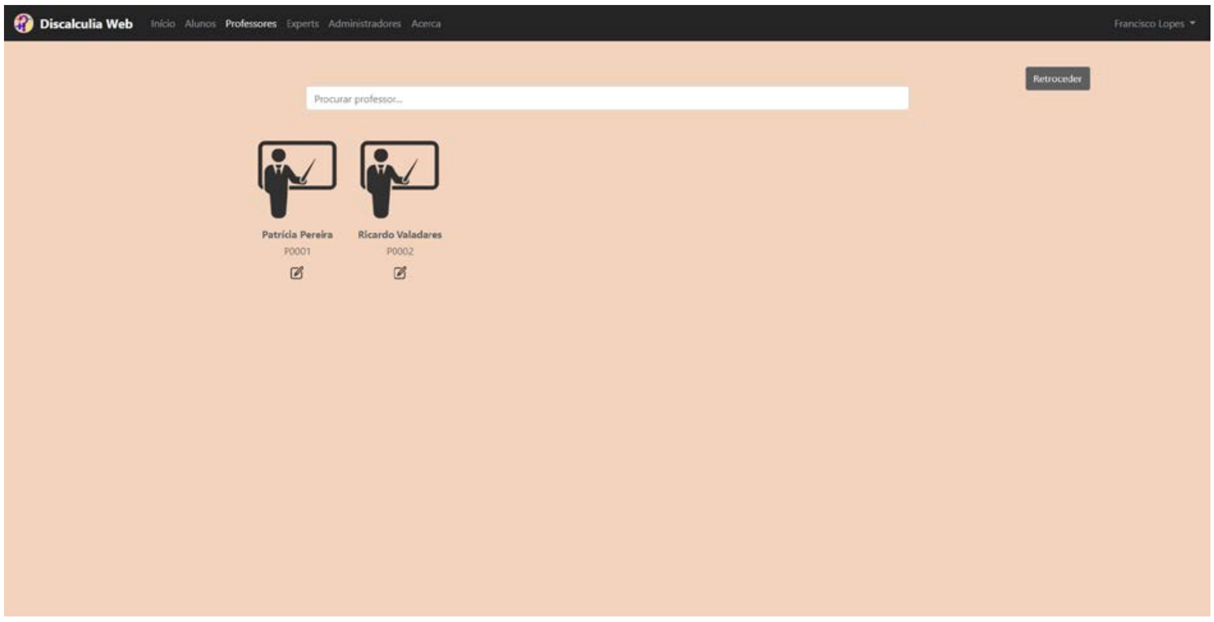


Figure 99 - Administrator teachers screen preview

In this screen, the administrator has access to all teachers and their view over the web application, by clicking on the teacher's image, as seen in the following figure (Figure 100):

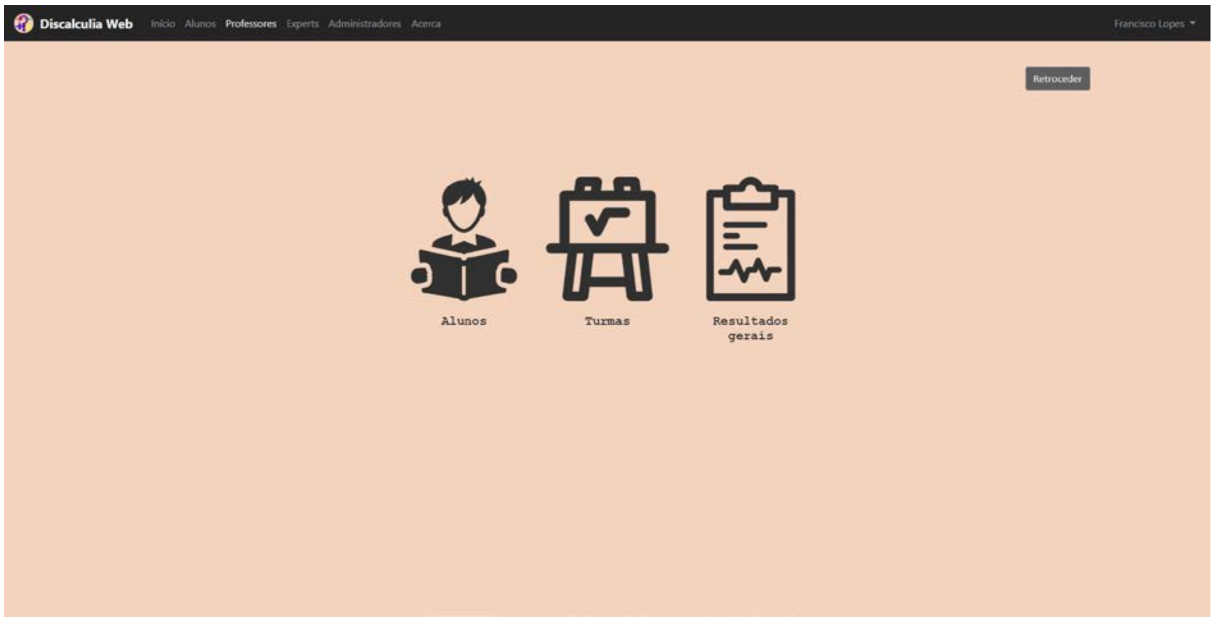
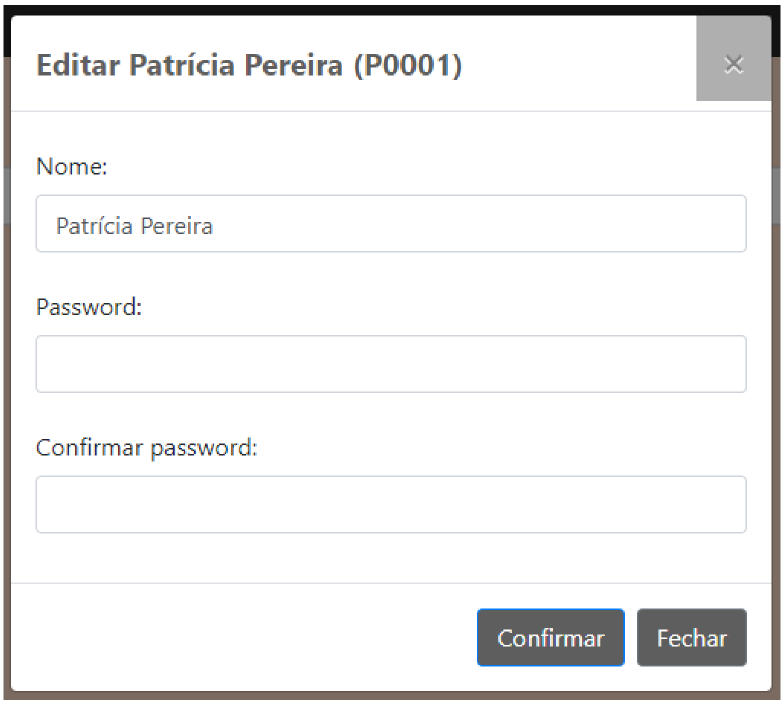


Figure 100 - Administrator teacher screen preview

The user also has the ability to edit the teacher's details by clicking on the image below the teacher's details. By doing this action, a pop up appears (Figure 101), as can be seen as follows:


Editar Patrícia Pereira (P0001)

Nome:

Patrícia Pereira

Password:

Confirmar password:

Figure 101 - Administrator teacher edition close-up

This pop-up loads and shows the selected teacher's name and two fields that allow the administrator to change the student's password, by typing it in the first box and repeating it in the second, an action that is useful in the case the teacher forgets the password.

Selecting the *Experts* option present on the initial administrator screen (Figure 92) or on the top left of the screen (Figure 93), the administrator is given access to the following page (Figure 102):

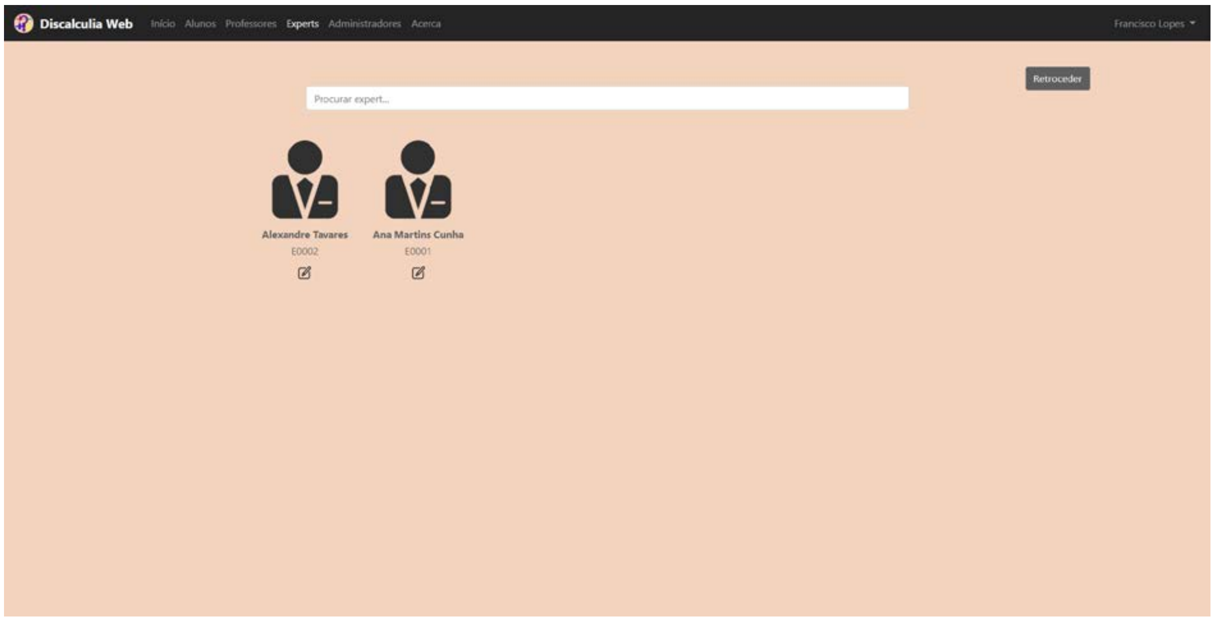


Figure 102 - Administrator experts screen preview

In this screen, the administrator has access to all experts and their screens. This can be done by clicking on the expert's image, as follows (Figure 103):

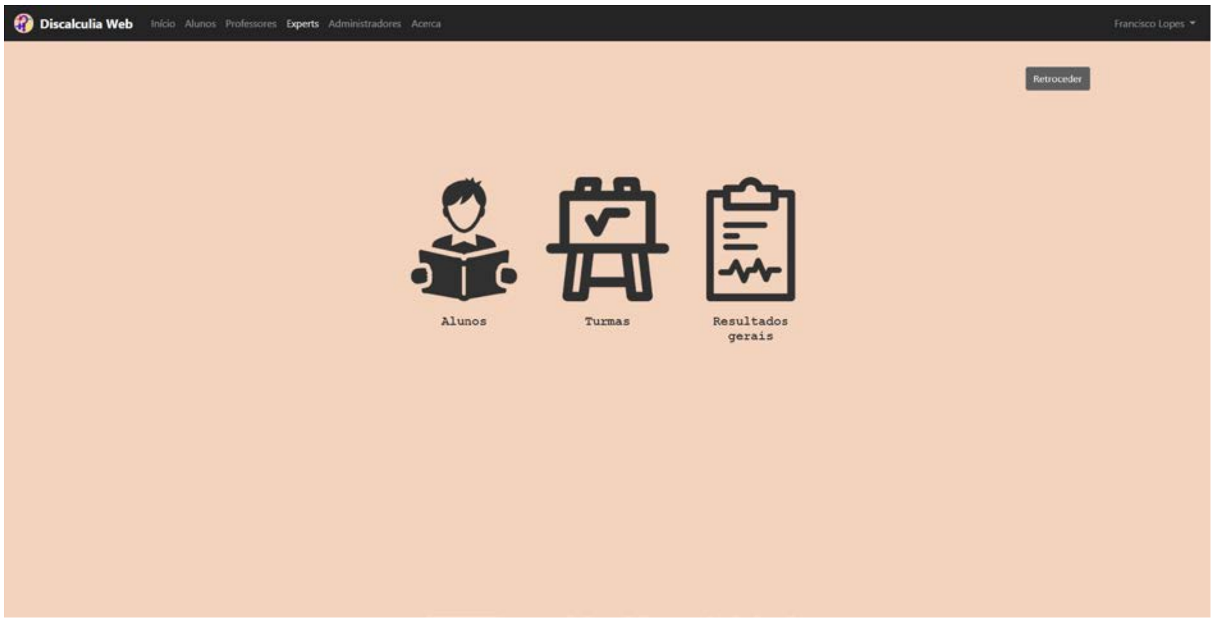
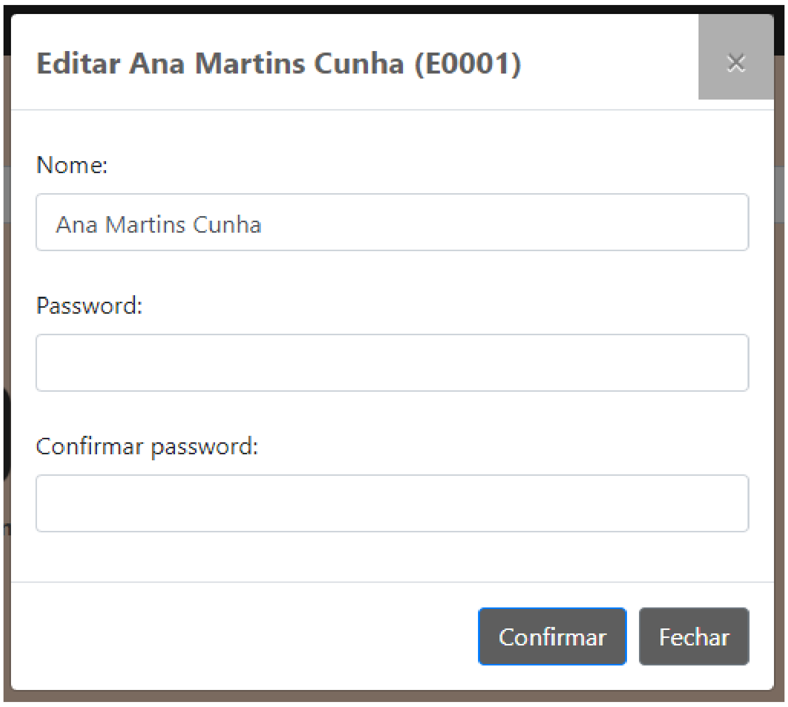


Figure 103 - Administrator expert screen preview

The user also has the ability to edit the expert's details, just like with the teacher's case (Figure 101) by clicking on the image below the expert's details, indicative of that possibility. By clicking it, the user makes a pop up appear, as can be seen in the following figure (Figure 104):


Editar Ana Martins Cunha (E0001)

Nome:

Ana Martins Cunha

Password:

Confirmar password:

Figure 104 - Administrator expert edition close-up

This pop-up loads the selected expert's name, allowing the possibility to change it, and two fields that allow the administrator to change the expert's password, by typing it twice in the right input boxes.

If the user selects the *Administradores* option present on the initial administrator screen (Figure 92) or on the top left of the screen (Figure 93), the web application presents the user the following page (Figure 105):

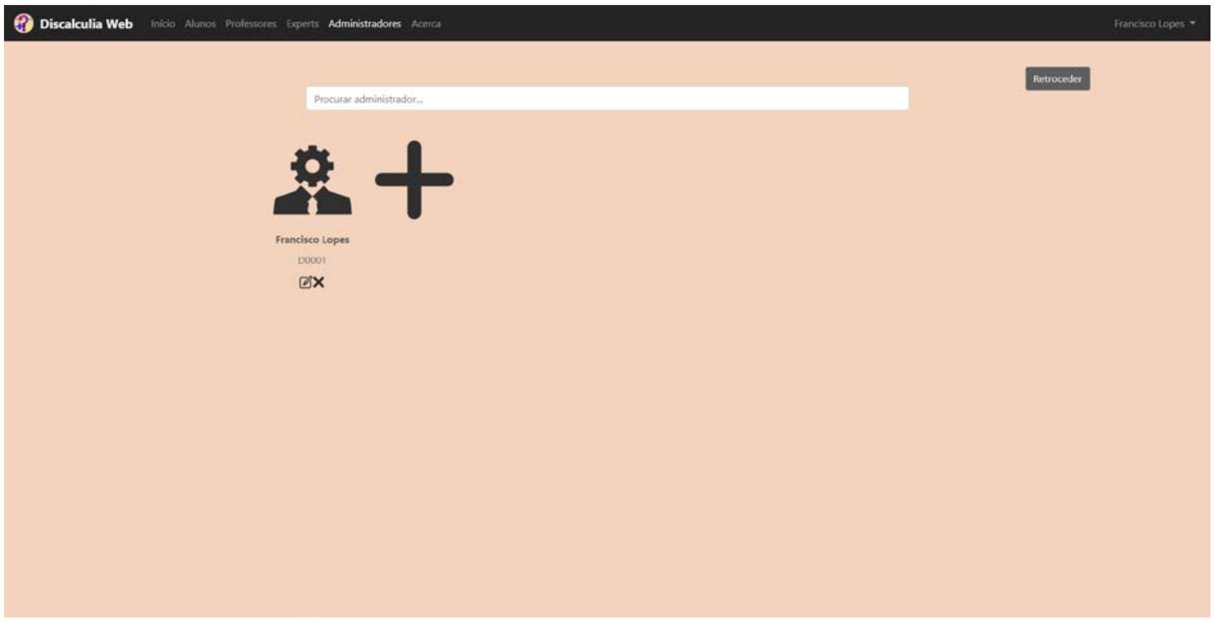


Figure 105 - Administrator administrators screen preview

Unlike the possibility that exists for the other user types, the user cannot gain access to the selected administrator's view over the web application, the image is purely cosmetic.

However, besides the user having the ability to edit the administrators' details by clicking on the respective image below the administrator's details, the user also has the ability to remove an administrator from the platform, again by clicking on the respective image below the administrator's details, and the ability to add a new administrator by clicking the plus sign next to last presented administrator.

If the user decides to edit an administrator's details, the following pop-up appears on screen (Figure 106):

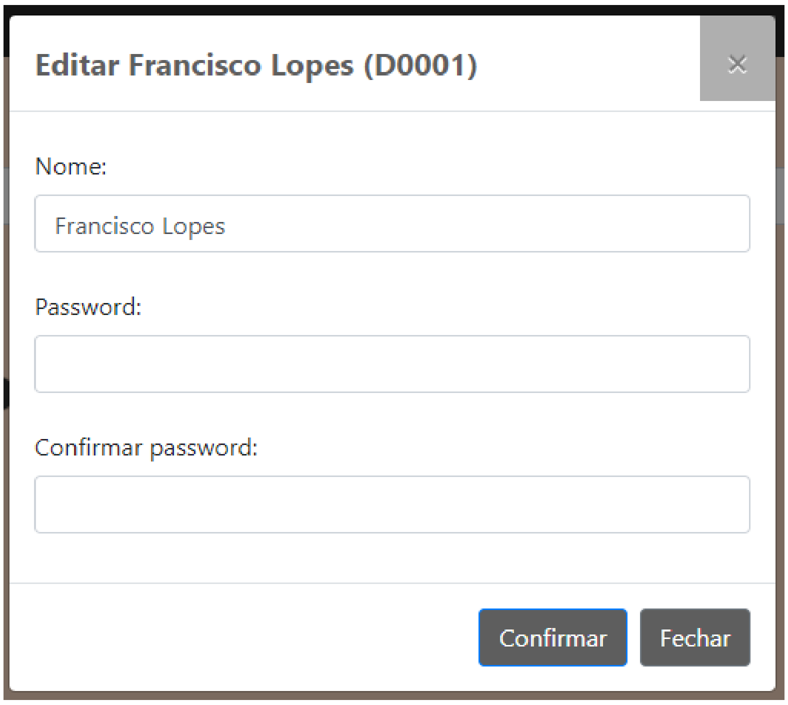
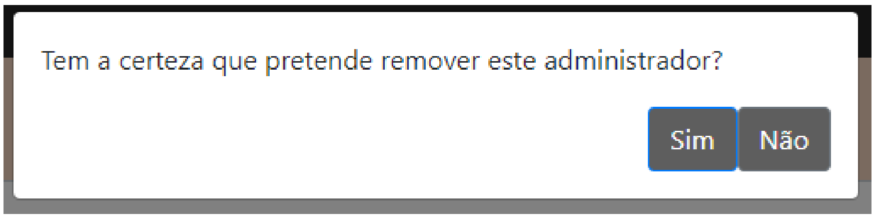


Figure 106 - Administrator administrator edition close-up

This pop-up loads and shows the selected administrator's name, allowing for the possibility to change it, and two fields that allow the user to change the selected administrator's password, by typing it in the first box and repeating correctly it in the second.

If the user decides to remove an administrator, a pop-up appears to confirm the action (Figure 107), and if the user intends to go through with the removal process, the administrator is removed from the web application system.



Tem a certeza que pretende remover este administrador?

Figure 107 - Administrator administrator removal pop-up

If the user decides to add an administrator, a pop-up appears to ask for the new administrator's name in order to add the new administrator to the platform's database with the proper details. This pop-up can be seen in the following figure (Figure 108):

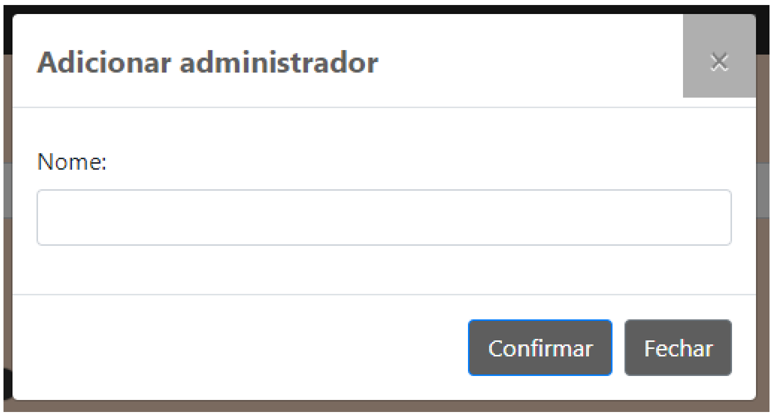


Figure 108 - Administrator administrator addition pop-up

These last figures constitute the final screens available to a user that accesses the web application. While the design and functionalities of the web application are discussed there is a remaining detail that must be discussed.

Since most pages require a user to be logged in into the web application, they cannot be made possible to be accessed by anyone, for example, by only typing the correct address and route. Due to Angular's capabilities, this process is simple.

Angular allows, when defining the routes and the pages that are shown, to set conditions for when those pages to be shown and loaded, two different actions that are both important. If these conditions are not set a user can access any page, possibly creating problems to the page's normal functioning, and load all pages just by connecting to the initial page, for example, if a user connects to the web application, the components responsible for the student's side of the application are also loaded, even if they are not used. This last can create further workload to web server that is providing the web application to the user, as well as create security issues, which can be even more worrying.

As such, every page that is shown and loaded only does so if the right conditions are met. The conditions that were set for this project include the verification of a valid login and the verification of the correct logged in user type, depending on the page.

At this point in the development process of the project, the web application was built with every planned function fully developed. As such, the next step was to host it, in order for it to be publicly available through the internet, using the respective software chosen for the purpose, a process that is discussed in the following section.

6.7. Hosting

With the web application fully developed and ready to be made public, the first step towards doing that was building the project. The building process is a necessary step since it reduces the size of the overall project substantially and makes the existing files as incomprehensible as possible. This was done by using tools available in the selected technologies that were mentioned in the previous step.

After that was done, the hosting process itself was very similar to the application programming interface since all that was needed was to inform the hosting software, Internet Information Services, that the folder that was built was the one that was expected to be hosted.

But, since the hosting software was running on an older version, like mentioned in the Application programming interface implementation section, there was a problem related to the routes that were implemented on the web application. That problem did not allow accessing any route directly except for the one that led to the first page since it did not have any route parameters. For example, if the address to the web application was http://webapplication.com and there were other routes like http://webapplication.com/student and http://webapplication.com/teacher, only the first address could be accessed and the last two presented the user with a blank page.

The solution to that problem was done by using a software program that implemented itself on the Internet Information Services hosting program, ISAPI_Rewrite. Since there wasn't a lot of information on the internet on how to solve this problem by using said software and how to use the program for the intended purpose of this project, the solution came through thanks to trial and error. All routes had to be manually configured but, after some tests, all routes were properly configured, and the web application and routes were properly accessible through a public address on the internet [112].

6.8. Testing

At this point, everything about the web application and all the other components of the project were developed and all available through a public address on the internet. All that was left was to ensure that everything was working together as expected on the web application.

The first tests were done during the development stage and was the first test that was done every single feature that was developed. This was possible due to Visual Studio Code, the software program used for coding the web application, which has support for a tool called Node.js which permitted to execute the code that was developed and test the features that were done.

When the web application was built and deployed all tests were focused on making sure every feature was still working as intended. Different scenarios were tested in order to ensure that every single opportunity regarding page and route navigation were working properly as well as other possible problems any feature could have were discovered and fixed.

Since this application was accessible through any modern web browser at the time of the development of this project, access through a smartphone was also possible and taken into account during the development phase. As such, when the web application was available publicly, all features and pages were tested through a smartphone and also through a tablet device, a process just like the one described in the previous paragraph.

Having done all this testing, and more due to unplanned spontaneous usage, there was no problem detected nor any change that was deemed necessary to ensure proper functioning of the web application, something that was majorly due to the possibility of testing during the development phase and the planning process that occurred.

7. Discussion and conclusion

7.1. Project analysis

In this section all the work that was put in the final result as well as the final result itself, including its usefulness, effectiveness and potential will be studied and discussed. The method that was used to describe these aspects is called SWOT analysis, where SWOT is an acronym for strengths, weaknesses, opportunities and threats.

A SWOT analysis is as analysis process that can be done to a project in order to determine, as its name implies, its strengths, weaknesses, opportunities and threats, so that it is possible to understand if the results and process were aligned with the objectives and goals that it set out to accomplish [113]–[115].

A projects strengths are constituted by its characteristics that can be considered positive or factors that are thought to be well done [113]–[115]. In the case of this project, the strengths that were compiled are as follows:

- The web application has a modern, simple and practical graphical interface, characteristics that that are mostly due to the use of state-of-the-art technologies;
- Strong planning phases that ensured a final product with all features that were needed with room for further development;
- Testing phases that ensured an error free web application, especially for regular users.

Its weakness are, obviously, the opposite of its strengths and are used to represent characteristics that can be seen as detrimental to the project's objectives or process [113]. As such, characteristics that are thought of as weaknesses of this project include:

- Although simplicity regarding data viewing is a goal it can also be considered a detriment to this application since most of its better-known competition are quite more data driven, presenting their data in more complex fashions;
- The platform may be considered a little confined to its environment since the use that may come of it can only be completely effective when used in a primary school setting (junior school in the United Kingdom or elementary school in the United States of America).

These two previous points of the SWOT analysis that is being discussed are constituted of factors that are considered to be due to the entity that is responsible for the development of the project. The next two points, opportunities and threats, are, on the contrary, attributed to external factors, for example, other projects or products with similar objectives, the market that exists for what is being developed, among others [113]–[115].

That being, the opportunities aspect of the SWOT analysis includes factors that can be responsible for the project's possible success [113]. In the case of this project, the opportunities include:

- Although there is a market for applications such as the one developed on this project, none was found that was directly involved with all entities around younger students in a school environment;
- In Portugal there are little to none known projects of this nature, a fact that makes this project unique and interesting, especially for schools around the country;
- Since mobile technologies are very popular around the world, if the games associated with this technology are successful, there will be considerable data to test and analyze, giving room for further improving the application.

Threats, on the other hand, are made of factors that have to be accounted for and have to be anticipated for since, in the long run, they can be detrimental to a project's strengths and add points to its weaknesses [113]–[115]. The list of threats that were considered in this project are as follows:

- This is a relatively simple project regarding data viewing which make other similar platforms that are more data oriented stronger in that aspect, since their data is shown though more complex and intricate means;
- Dyscalculia is not a very well-known disorder, especially when compared with its similar counterpart dyslexia, which can make it difficult to spread information about a platform such as this one.

This analysis allows for another layer of understanding of this project to be added and can be seen as the point of view on the situation a project is inserted in on in the eyes of the entity responsible for developing it, a point of view that is important to be described and be passed on via this document that aims to discuss every aspect of the project.

7.2. Conclusion and future work

Having developed all the features and components that were meant to be developed for this project, a few notes have been taken and are going to be discussed in this section as a means to conclude this dissertation, a document which primary focus is to explain the development process and the thinking process behind it.

All features that were planned at the start of the development process were fully developed and are all working as they were intended and with as much polish as it was possible, especially all features that are accessible to users with permission to make use of them. In that regard, this project was successful.

An interesting point regarding the development of each component was the adaptability there had to exist to different technologies, modern or old. Regarding the web application, the technology that it is based upon, Angular, is a modern technology with a very contemporary look and features, characteristics that had to work together with older technologies due to the resources that were available for developing the project, namely the hosting software and the application programming interface.

Some problems arose during this development project and several decisions had to be made. At the start of the database component development there had to be a change in which technology was going to be used, a change that came together with the reconstruction of the database and its repopulation, this last process mainly for testing purposes. Another problem that existed was due to the older software versions mentioned in the previous paragraph. When trying to host the application programming interface there were problems while trying to make different types of request in which they could not be answered. This problem was solved by testing and remaking the web application in different versions and adapting to the machine that was available. The last major problem, discussed in the Web application hosting section, did not allow the routes that existed on the web application to be accessible directly, a problem that was solved by using an older and relatively unknown and undocumented software in which said routes had to all been manually configured, a process that was slow and required trial and error in order to figure out. All these problems, and some other minor ones, where all figured out and resulted in an application that is more robust and less prone to problems later on in its life.

Another important factor to mention regarding this section is its possibility to be expanded to include support for other environments. As it was mentioned during this document, the focus of this project was, for the time, a single school, to which a visit was made and data was gathered, but, it is possible to expand this project to other schools, an expansion that can serve as possible future work that can be done upon this project, as well as add more features that can enrich the data viewing capabilities of the web application.

To conclude, although this project was extensive and there were problems during its development process, the challenge it posed was beat and the goals that were set were accomplished with room for further development and addition of several features, all in the name of diagnosing and treating dyscalculia in order to make the world a little better.

References

[1] F. Ferraz and J. Neves, "A Brief Look Into Dyscalculia And Supportive Tools," 2015.

[2] the free encyclopedia From Wikipedia, "Dyscalculia," 2005. [Online]. Available: https://en.wikipedia.org/wiki/Dyscalculia.

[3] the F. E. From Wikipedia, "Dyslexia," 2003. [Online]. Available: https://en.wikipedia.org/wiki/Dyslexia.

[4] M. M. Ariffin, F. A. A. Halim, N. Abd, and Aziz, "MOBILE APPLICATION FOR DYSCALCULIA CHILDREN IN MALAYSIA," 2017.

[5] "Dyscalculia: Definition, Types, Symptoms, Incidence, Causes and Program." [Online]. Available: https://www.edubloxtutor.com/dyscalculia-definition-types-symptoms-incidence-causes/.

[6] S. Bhandari, "What Is Dyscalculia? What Should I Do if My Child Has It?," 2017. [Online]. Available: https://www.webmd.com/add-adhd/childhood-adhd/dyscalculia-facts.

[7] the free encyclopedia From Wikipedia, "Neuroplasticity," 2007. [Online]. Available: https://en.wikipedia.org/wiki/Neuroplasticity.

[8] M. Singh, "Dyscalculia treatment – How to effectively build math skills?," 2018. [Online]. Available: http://numberdyslexia.com/dyscalculia-treatment/.

[9] the F. E. From Wikipedia, "Cloud database," 2012. [Online]. Available: https://en.wikipedia.org/wiki/Cloud_database.

[10] the free encyclopedia From Wikipedia, "Microsoft SQL Server," 2004. [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_SQL_Server.

[11] M. Rouse, "Microsoft SQL Server." [Online]. Available: https://searchsqlserver.techtarget.com/definition/SQL-Server.

[12] the free encyclopedia From Wikipedia, "Relational database," 2005. [Online]. Available: https://en.wikipedia.org/wiki/Relational_database.

[13] "Relational Model." [Online]. Available: https://www.techopedia.com/definition/24559/relational-model-database.

[14] the free encyclopedia From Wikipedia, "SQL," 2003. [Online]. Available: https://en.wikipedia.org/wiki/SQL.

[15] "What is SQL?" [Online]. Available: http://www.sqlcourse.com/intro.html.

[16] A. Stefanuk, "A beginner's guide to SQL," 2018. [Online]. Available: https://learntocodewith.me/posts/sql-guide/.

[17] Z. Allen, "Benefits of Microsoft SQL Server." [Online]. Available: https://help.acctivate.com/articles/5914/.

[18] the F. E. From Wikipedia, "SQL Server Management Studio," 2011. [Online]. Available: https://en.wikipedia.org/wiki/SQL_Server_Management_Studio.

[19] Ian, "What Is SQL Server Management Studio?," 2016. [Online]. Available: https://database.guide/what-is-sql-server-management-studio/.

[20] M. Rouse, "Query." [Online]. Available: https://searchsqlserver.techtarget.com/definition/query.

[21] C. Hoffman, "What Is Microsoft Azure, Anyway?," 2018. [Online]. Available: https://www.howtogeek.com/337961/what-is-microsoft-azure/.

[22] the F. E. From Wikipedia, "Microsoft Azure," 2009. [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Azure.

[23] Microsoft, "O que é o Azure?" [Online]. Available: https://azure.microsoft.com/pt-pt/overview/what-is-azure/.

[24] Microsoft, "Base de Dados SQL." [Online]. Available: https://azure.microsoft.com/pt-pt/services/sql-database/.

[25] the F. E. From Wikipedia, "Microsoft Azure SQL Database," 2011. [Online]. Available:

https://en.wikipedia.org/wiki/Microsoft_Azure_SQL_Database.
[26] the F. E. From Wikipedia, "Firebase," 2014. [Online]. Available: https://en.wikipedia.org/wiki/Firebase.
[27] GeekyAnts, "Introduction to Firebase," 2017. [Online]. Available: https://hackernoon.com/introduction-to-firebase-218a23186cd7.
[28] "Firebase Realtime Database." [Online]. Available: https://firebase.google.com/docs/database/.
[29] "Understand Firebase Realtime Database Rules." [Online]. Available: https://firebase.google.com/docs/database/security/.
[30] "Learn What Azure SQL Databases and Microsoft Access Can Do for You!" [Online]. Available: http://gainingaccess.net/SQLAzure/AccessAndSQLAzureBE.aspx.
[31] "What Are the Pros and Cons of Microsoft Azure?," 2017.
[32] Jburks725, "Firebase pros and cons," 2014. [Online]. Available: https://gist.github.com/jburks725/1ad9d8a14a1351941bc8.
[33] A. Dey, "No Title," 2016. [Online]. Available: https://www.quora.com/What-are-the-disadvantages-of-using-Firebase-as-backend-for-mobile-app.
[34] L. Nathan, "Firebase Pros and Cons," 2016. [Online]. Available: https://medium.com/one-tap-software/firebase-pros-and-cons-ce37c766190a.
[35] ruffrey, "Firebase Pros and Cons for a SaaS Company," 2018. [Online]. Available: https://symboliclogic.io/firebase-pros-and-cons/.
[36] M. Rouse, "data analytics (DA)," 2019. [Online]. Available: https://searchdatamanagement.techtarget.com/definition/data-analytics.
[37] S. Arora, "Top 14 Areas for Data Analytics Application," 2017. [Online]. Available: https://www.digitalvidya.com/blog/data-analytics-applications/.
[38] M. Lebied, "How to Build Interesting Data Reports People Love to Read," 2016. [Online]. Available: https://www.datapine.com/blog/data-reports-people-love-to-read/.
[39] "What is a Dashboard?" [Online]. Available: https://www.idashboards.com/guides/what-is-a-dashboard/.
[40] the F. E. from wikipedia, "Power BI," 2017. [Online]. Available: https://en.wikipedia.org/wiki/Power_BI.
[41] M. P. BI, "Power BI Embedded analytics | Microsoft Azure." [Online]. Available: https://azure.microsoft.com/en-us/services/power-bi-embedded/.
[42] "Microsoft Power BI Features." [Online]. Available: https://www.getapp.com/business-intelligence-analytics-software/a/microsoft-power-bi/features/.
[43] the F. E. From Wikipedia, "Elasticsearch," 2012. [Online]. Available: https://en.wikipedia.org/wiki/Elasticsearch.
[44] "Elasticsearch." [Online]. Available: https://aws.amazon.com/elasticsearch-service/what-is-elasticsearch/.
[45] the F. E. From Wikipedia, "Kibana," 2018. [Online]. Available: https://en.wikipedia.org/wiki/Kibana.
[46] "Kibana." [Online]. Available: https://aws.amazon.com/pt/elasticsearch-service/kibana/.
[47] "Kibana." [Online]. Available: https://www.elastic.co/products/kibana.
[48] "Kibana features." [Online]. Available: https://www.elastic.co/products/kibana/features.
[49] "10 Advantages of Microsoft Power BI," 2017. [Online]. Available: https://www.snp.com/blog/10-advantages-microsoft-power-bi.
[50] Microsoft, "Pricing." [Online]. Available: https://powerbi.microsoft.com/en-us/pricing/.
[51] T. E. Team, "Top 10 Benefits of Leveraging Power BI for Reporting Sales in Dynamics NAV," 2017. [Online]. Available: https://www.encorebusiness.com/blog/power-bi-top-10-benefits/.
[52] D. Team, "8 Important Power BI Advantages and Disadvantages – 2018," 2018. [Online].

Available: https://data-flair.training/blogs/power-bi-advantages-and-disadvantages/.
[53] "Microsoft Power BI Pros and Cons." [Online]. Available: https://www.absentdata.com/power-bi-pros-and-cons/.
[54] "Elasticsearch Tutorial." [Online]. Available: https://www.tutorialspoint.com/elasticsearch.
[55] C. Vig, "Elasticsearch: Indexing SQL databases. The easy way.," 2014. [Online]. Available: http://blog.comperiosearch.com/blog/2014/01/30/elasticsearch-indexing-sql-databases-the-easy-way/.
[56] "Elasticsearch Pros and Cons | Advantages and Disadvantages of Elasticsearch," 2018. [Online]. Available: https://interviewbubble.com/elasticsearch-pros-and-cons-advantages-and-disadvantages-of-elasticsearch/.
[57] Insights, "The 3 reasons why you should be using ELK," 2016. [Online]. Available: https://www.mcplusa.com/the-3-reasons-why-you-should-be-using-elk/.
[58] the F. E. From Wikipedia, "Angular (application platform)," 2017. [Online]. Available: https://en.wikipedia.org/wiki/Angular_(application_platform).
[59] "What is Angular?" [Online]. Available: https://angular.io/docs.
[60] Ravi, "Getting Started with Angular 2 using TypeScript," 2015. [Online]. Available: https://www.sitepoint.com/getting-started-with-angular-2-using-typescript/.
[61] "Typescript vs JavaScript: What's the Difference?" [Online]. Available: https://www.guru99.com/typescript-vs-javascript.html.
[62] the F. E. From Wikipedia, "Scripting language," 2004. [Online]. Available: https://en.wikipedia.org/wiki/Scripting_language.
[63] D. López, "My favorite Angular libraries to use in any project," 2019. [Online]. Available: https://dev.to/frostqui/my-favorite-angular-libraries-to-use-in-any-project-47ai.
[64] "Features & benefits." [Online]. Available: https://angular.io/features.
[65] "Benefits of Angular Framework To Develop Modern Applications," 2019. [Online]. Available: https://www.rishabhsoft.com/blog/reasons-to-choose-angular-for-application-development.
[66] "Getting Started." [Online]. Available: https://code.visualstudio.com/docs.
[67] the F. E. From Wikipedia, "Visual Studio Code," 2015. [Online]. Available: https://en.wikipedia.org/wiki/Visual_Studio_Code.
[68] Anonymous, "About | Node.js," 2016. [Online]. Available: https://nodejs.org/en/about/.
[69] the F. E. From Wikipedia, "Node.js," 2011. [Online]. Available: https://en.wikipedia.org/wiki/Node.js.
[70] B. Cameron, "7 Essential Features of Visual Studio Code for Web Developers," 2019. [Online]. Available: https://medium.com/@bretcameron/7-essential-features-of-visual-studio-code-for-web-developers-be77e235bf62.
[71] B. Holland and S. Drasner, "VS Code can do that?!" [Online]. Available: https://vscodecandothat.com/.
[72] "Pencil Project." [Online]. Available: https://pencil.evolus.vn/.
[73] A. Siddiqui, "What is mockup?," 2016. [Online]. Available: https://www.quora.com/What-is-mockup.
[74] infotechaliveadmin, "Why a Mockup is Needed Before Starting a Design or Development," 2018. [Online]. Available: http://www.infotechalive.com/why-a-mockup-is-needed-before-starting-a-design-or-development/.
[75] "Top features of Pencil." [Online]. Available: https://pencil.evolus.vn/Features.html.
[76] the F. E. From Wikipedia, "ASP.NET," 2005. [Online]. Available: https://en.wikipedia.org/wiki/ASP.NET.
[77] the F. E. From Wikipedia, "Web framework," 2007. [Online]. Available: https://en.wikipedia.org/wiki/Web_framework.

[78] the F. E. From Wikipedia, “Web service,” 2004. [Online]. Available: https://en.wikipedia.org/wiki/Web_service.

[79] M. Watson, “Top 13 ASP.NET Core Features You Need to Know,” 2017. [Online]. Available: https://stackify.com/asp-net-core-features/.

[80] the free encyclopedia From Wikipedia, “Microsoft Visual Studio,” 2004. [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio.

[81] the free encyclopedia From Wikipedia, “Integrated development environment,” 2004. [Online]. Available: https://en.wikipedia.org/wiki/Integrated_development_environment.

[82] the F. E. from wikipedia, “Internet Information Services,” 2005. [Online]. Available: https://en.wikipedia.org/wiki/Internet_Information_Services.

[83] Microsoft, “IIS.” [Online]. Available: https://www.iis.net/overview.

[84] R. Machado, “Desenvolver Banco de Dados em 7 etapas,” 2016. [Online]. Available: http://fluxoconsultoria.poli.ufrj.br/blog/tecnologia-informacao/desenvolvimento-banco-de-dados/. [Accessed: 07-Apr-2019].

[85] M. R. Blaha, “Building Database Applications,” 2001. [Online]. Available: http://www.informit.com/articles/article.aspx?p=21702&seqNum=6. [Accessed: 07-Apr-2019].

[86] S. Nicholson, “Building a Database for Dynamic Applications in Macromedia DreamWeaver MX 2004,” 2004. [Online]. Available: http://www.peachpit.com/articles/article.aspx?p=102302&seqNum=2. [Accessed: 07-Apr-2019].

[87] the free encyclopedia From Wikipedia, “Conceptual model,” 2010. [Online]. Available: https://en.wikipedia.org/wiki/Conceptual_model.

[88] A. Powell-Morse, “Conceptual Models – What Are They and How Can You Use them?,” 2017. [Online]. Available: https://airbrake.io/blog/sdlc/conceptual-model.

[89] G. Hineman, “What Is the Difference Between a Conceptual Model & A Physical Model?,” 2017. [Online]. Available: https://www.theclassroom.com/what-is-the-difference-between-a-conceptual-model-a-physical-model-12592752.html.

[90] S. Gupta, “Difference between Conceptual, Logical and Physical Data Models,” 2012. [Online]. Available: https://uksanjay.blogspot.com/2012/06/difference-between-conceptual-logical.html.

[91] “Discover the Benefits of Visual Paradigm.” [Online]. Available: https://www.visual-paradigm.com/features/.

[92] the free encyclopedia From Wikipedia, “Class diagram,” 2006. [Online]. Available: https://en.wikipedia.org/wiki/Class_diagram.

[93] “Many-to-many relationships.” [Online]. Available: https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP_Help/many-to-many-relationships.html.

[94] L. Li, “Database Normalization Explained,” 2019. [Online]. Available: https://towardsdatascience.com/database-normalization-explained-53e60a494495.

[95] “What is Normalization? 1NF, 2NF, 3NF & BCNF with Examples.” [Online]. Available: https://www.guru99.com/database-normalization.html.

[96] IBM, “Normalize a data model.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSGU8G_14.1.0/com.ibm.ddi.doc/ids_ddi_066.htm.

[97] the free encyclopedia From Wikipedia, “Third normal form,” 2007. [Online]. Available: https://en.wikipedia.org/wiki/Third_normal_form.

[98] “SQL Constraints.” [Online]. Available: https://www.w3schools.com/sql/sql_constraints.asp.

[99] M. Filina, “The Big Bad Guide on Database Testing,” 2019. [Online]. Available: https://hackernoon.com/database-testing-the-guide-wq30i300n.

[100] “SQL Stored Procedures for SQL Server.” [Online]. Available: https://www.w3schools.com/sql/sql_stored_procedures.asp.
[101] the free encyclopedia From Wikipedia, “Database trigger,” 2006. [Online]. Available: https://en.wikipedia.org/wiki/Database_trigger.
[102] T. E. Bettilyon, “What Is an API and Why Should I Use One?,” 2018. [Online]. Available: https://medium.com/@TebbaVonMathenstien/what-is-an-api-and-why-should-i-use-one-863c3365726b.
[103] “Use Case Diagram.” [Online]. Available: https://www.smartdraw.com/use-case-diagram/.
[104] “What is Use Case Diagram?” [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/.
[105] “HTTP Requests.” [Online]. Available: https://www.codecademy.com/articles/http-requests.
[106] “Postman | The Collaboration Platform for API Development.” [Online]. Available: https://www.getpostman.com/.
[107] the F. E. From Wikipedia, “Radar chart,” 2010. [Online]. Available: https://en.wikipedia.org/wiki/Radar_chart.
[108] C. Hope, “Framework,” 2017. [Online]. Available: https://www.computerhope.com/jargon/f/framework.htm.
[109] N. Choudhary, “What is a software framework?,” 2012. [Online]. Available: https://stackoverflow.com/questions/2964140/what-is-a-software-framework.
[110] “Bootstrap.” [Online]. Available: https://getbootstrap.com/.
[111] “Materialize.” [Online]. Available: https://materializecss.com/.
[112] “ISAPI_Rewrite 3 - Apache .htaccess mod_rewrite compatible module for IIS.” .
[113] N. Parsons, “What Is a SWOT Analysis, and How to Do It Right (With Examples),” 2018. [Online]. Available: https://www.liveplan.com/blog/what-is-a-swot-analysis-and-how-to-do-it-right-with-examples/.
[114] the free encyclopedia From Wikipedia, “SWOT analysis,” 2005. [Online]. Available: https://en.wikipedia.org/wiki/SWOT_analysis.
[115] D. Shewan, “How to Do a SWOT Analysis for Your Small Business (with Examples),” 2019. [Online]. Available: https://www.wordstream.com/blog/ws/2017/12/20/swot-analysis.