INTELI - INSTITUTO DE TECNOLOGIA E LIDERANÇA
SOFTWARE ENGINEERING

BRUNO OTAVIO BEZERRA DE MEIRA

INTERACTIVE VIRTUAL ENVIRONMENT FOR SECONDARY SCHOOLS: APPLYING VIRTUAL REALITY TO THE TEACHING-LEARNING PROCESS

SÃO PAULO
2025

BRUNO OTAVIO BEZERRA DE MEIRA

INTERACTIVE VIRTUAL ENVIRONMENT FOR SECONDARY SCHOOLS: APPLYING VIRTUAL REALITY TO THE TEACHING-LEARNING PROCESS

Course Conclusion Final presented to the Faculty of the Software Engineering course at the Institute of Technology and Leadership, as part of the requirements for obtaining a Bachelor's Degree in Software Engineering.

Advisor:
Prof. Msc. Murilo Zanini de Carvalho

São Paulo, SP
2025

CATALOG SHEET

Dedication: To-Do

ACKNOWLEDGMENTS

To-do

“Epigraph:TO-DO”
Autor

ABSTRACT

To-do

Keywords: Keywords.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BNCC	Brazilian National Common Curricular Base
VR	Virtual Reality
AR	Augmented Reality
3D	Three Dimensional
TAM	Technology Acceptance Model
ICT	Information and Communication Technologies
BMC	Business Model Canvas

SUMMARY

1 INTRODUCTION

We are living in an era marked by significant technological advances, which have profoundly transformed various aspects of society, including the educational field. The growing integration of technology into people's daily lives requires us to reflect on the possibilities of its application in the teaching-learning process, especially regarding the potential improvement of traditional methodologies used in the classroom. This transformation aims not only to optimize the teacher's pedagogical practices but also to promote student engagement and motivation in relation to the curricular content established by the Brazilian National Common Curricular Base (BNCC). Various theorists, such as Freire (1996), have long advocated overcoming a banking, traditional, and passive education model, proposing instead a student-centered approach based on dialogue, active participation, and the stimulation of knowledge construction.

In this context, it is observed that purely traditional teaching methods no longer effectively meet contemporary educational demands. This is where the concept of Active Learning emerges, proposing student-centered pedagogical strategies that emphasize problem-solving, hypothesis formulation, and active participation in the construction of knowledge (ROCHA; LEMOS, 2014).

This demand for new educational strategies has paved the way for innovative approaches capable of making the teaching-learning process more dynamic and appealing to students. These approaches involve the integration of technological resources that support the transformation of pedagogical practices, enabling new forms of interaction and knowledge building. Among these resources, Virtual Reality (VR) and Augmented Reality (AR) stand out.

Virtual Reality is an advanced interface for computer applications that allows users to navigate and interact in real time within a three-dimensional environment, often using multisensory devices for action or feedback (Tori, Kirner, and Siscoutto, 2006). This technology enables interactive and immersive representations of both imaginary scenarios and real-world reproductions, facilitating user engagement in a 3D space. In addition to conventional interaction via menus or buttons, VR allows individuals to directly manipulate three-dimensional elements, enhancing their perceptual and sensory capabilities in both time and space. Through 3D modeling, virtual objects and scenarios are developed, enabling a richer and more immersive navigation experience.

The advancement of multimedia technologies and Virtual Reality has been driven by the increasing processing power of computers, allowing real-time integration between videos and interactive virtual environments. Augmented Reality, in turn, has directly benefited from these advances by enabling the overlay of virtual elements onto the physical environment, expanding its potential applications in various contexts. Its use has become feasible not only on sophisticated platforms but also on accessible devices such as smartphones, democratizing its reach. Unlike Virtual Reality, which completely replaces the real environment with a simulated one, Augmented Reality inserts virtual objects into the user's physical space, fostering a more fluid, intuitive, and accessible interaction experience—often eliminating the need for training or complex equipment (TORI; KIRNER; SISCOUTTO, 2006). Studies such as that by Macedo and Fernandes (2015) highlight that the use of AR in educational contexts provides advantages such as 3D visualization of experiments, opportunities for students to interact with content, and simplicity in the use of the technology, making it a viable and efficient learning tool.

Considering, therefore, the continuous evolution of technological resources and the need for more effective and engaging teaching methodologies, this study analyzes the use of Augmented Reality and Virtual Reality as innovative and promising tools to enhance classroom learning. These digital and interactive technologies are becoming established as important pedagogical resources, not only for the benefits they offer to the teaching-learning process but also due to the urgent need to align educational practices with the technological transformations shaping the contemporary world.

This work is structured as follows: Section 2 presents a market study that aims to assess the feasibility and potential of AR and VR technologies in the school environment; Sections 3 and 4 describe the planning and development of the proposed solution, respectively; Section 5 discusses the tests conducted in an educational context, along with the analysis of the results and the perceived impacts on student learning; finally, Section 6 offers concluding remarks, summarizing the contributions and reflections generated throughout the study.

1.1 PROBLEM

In recent decades, education has faced significant challenges, particularly with regard to student engagement. As highlighted by Pozo (2002), one of the main obstacles encountered by educators is the lack of interest and motivation among students—a factor that directly compromises the teaching-learning process. Motivation, according to the author, is associated with psychological processes that influence student behavior in learning situations, and is therefore an essential element to be considered in pedagogical practice. Traditional methods, mostly developed in pre-digital contexts, have increasingly proven insufficient to meet the pedagogical demands of a generation that is growing up immersed in digital technologies (PRENSKY, 2001). This gap between teaching practices and student expectations contributes to demotivation and the consequent decline in academic performance.

Moreover, the Brazilian National Common Curricular Base emphasizes the importance of adopting methodologies that foster the development of socio-emotional competencies and practical skills—aspects that are often not fully addressed in traditional teaching approaches.

Although the benefits of using educational technologies are widely recognized, their implementation in Brazilian school contexts is still at an early stage. Personalized learning, for example, has shown promise, but its implementation faces structural limitations, requiring from teachers not only technical proficiency but also additional time and dedication to identify and address the individual needs of their students (PETERSON et al., 2018). In light of this, several guiding questions arise:

a) How can the creation of interactive virtual environments based on VR and AR impact the teaching-learning process in the context of high school education? With an emphasis on enhancing didactic materials and promoting more engaging methodologies aligned with the technological and pedagogical transformations of the 21st century.
b) What existing applications of VR and AR in education, and what have their outcomes been?

c) Which subjects present greater feasibility for the implementation of these technologies? Are virtual and augmented reality applicable to all areas of knowledge?
d) How do students and teachers perceive and accept the adoption of new tools that complement traditional teaching, especially in subjects that require visual and experiential learning?
e) Do educational institutions have the necessary human, technical, and financial resources to support the implementation and maintenance of VR and AR-based platforms?

1.2 JUSTIFICATION

The relevance of this study is grounded in the growing urgency to overcome the challenges imposed by traditional education, especially in a scenario marked by constant technological evolution. The specialized literature has highlighted the transformative role of emerging technologies, such as Virtual Reality, which are configured not only as innovative tools within the pedagogical context but also as effective mechanisms for promoting meaningful student engagement. In this regard, Freitas (2020) emphasizes that VR has been consolidating itself as a promising technology for overcoming physical barriers to communication and experimentation, expanding learning possibilities through immersive and interactive experiences. This perspective underscores the potential of VR to foster active learning practices and to facilitate the assimilation of complex content, contributing to the construction of deeper and more lasting knowledge.

From a practical standpoint, the integration of VR into the high school context proves to be a relevant strategy for the development of essential student skills, such as critical thinking, creativity, problem-solving, and collaborative work—competencies aligned with the guidelines established by the Brazilian National Common Curricular Base. Although immersive technologies are frequently applied in fields that require a high degree of visualization, such as Biology, Geography, and History, their potential extends beyond the boundaries of specific subjects. As noted by SENAI (2019, p. 37), collaborative learning environments can also be significantly enhanced by this innovation, allowing for simultaneous interaction among different users and the integration of multiple sources of information, thereby enriching the educational process in a meaningful way.

Thus, this study is justified by the possibility of contributing to the democratization of access to educational technology in Brazil, proposing viable and scalable solutions for the application of Virtual Reality in public educational institutions. Such an initiative not only promotes digital inclusion but also positions Brazilian education in alignment with contemporary demands, equipping students to face the challenges of an increasingly technological and interconnected society.

1.3 OBJETIVES

1.3.1 GENERAL OBJECTIVE

Analyze the application of VR and AR as pedagogical tools in high school education, as well as to develop a prototype of an interactive environment tailored to the school context, assessing its potential to transform traditional methodologies and promote greater student engagement, in alignment with the competencies established by the Brazilian National Common Curricular Base.

1.3.2 SPECIFIC OBJECTIVE

a) To survey and map the main applications of VR and AR in education, focusing on case studies, practical experiences, and outcomes achieved.
b) To investigate the potential of Virtual Reality as a tool for interactive learning, identifying its contributions to increased student engagement, improved content comprehension, enhanced motivation, knowledge retention, and learning outcomes.
c) To identify high school subjects with the greatest adherence to the use of immersive technologies, considering their visual, experimental, and conceptual characteristics.
d) To analyze the perceptions of teachers and students regarding the integration of VR and AR as complementary tools to conventional teaching.
e) To assess the technical and operational feasibility of implementing these technologies in school institutions, taking into account aspects such as infrastructure, costs involved, and the need for teacher training.
f) To develop a prototype of an interactive virtual environment based on AR and/or VR aimed at the school context, aligning it with the BNCC guidelines and contemporary educational demands.
g) To carry out practical tests with the prototype in real school environments and analyze the data obtained from user interactions.
h) To verify the technical and financial feasibility of applying immersive technologies in the context of Brazilian public schools.

1.4 PREMISES

a) The adoption of immersive technologies, such as VR and AR, can enhance active learning and student motivation, as pointed out by recent literature.
b) The BNCC establishes the need for the development of cognitive and socio-emotional competencies that can be fostered through innovative methodologies, including the use of VR and AR.
c) The effective implementation of VR and AR in the school environment is conditioned by the overcoming of structural challenges, such as the availability of technological resources and teacher training.
d) Subjects involving three-dimensional modeling, simulation of phenomena, or the reconstruction of historical and geographical contexts present greater compatibility with immersive technologies.
e) The acceptance and adaptation of educational actors (teachers and students) are determining factors for the successful integration of these technologies.

1.5 RESTRICTIONS

a) The dependence on specific hardware (e.g., VR headsets, motion sensors) and compatible software may limit accessibility in under-resourced schools.
b) The costs associated with the acquisition, maintenance, and upgrading of VR and AR equipment represent a significant barrier for public institutions.
c) The scope of the research is limited by the available timeframe for data collection and experimentation in real educational environments.
d) The lack of familiarity among part of the teaching staff with immersive technologies requires investment in training, which may not be feasible in the short term.
e) The generalization of results may be limited by the heterogeneity of school contexts in Brazil, requiring additional studies for large-scale validation.
f) Scope restricted to one or a few schools for testing and validation, due to project logistics and scheduling.
g) Consideration of ethical and legal aspects related to data privacy, student participation, and content moderation to ensure compliance with educational regulations.
h) VR technology may present usability challenges, such as motion sickness or difficulty in navigating virtual environments, which may affect user experience.
i) The prototype development will prioritize the use of open-source tools and self-hosted solutions to avoid technical or commercial barriers and to ensure greater autonomy and customization in the application of the solution.

1.6 DELIMITATION OF THE OBJECTIVE

This project is limited to the development of a solution based on Virtual Reality and/or Augmented Reality, with the objective of enhancing the teaching-learning process in schools, focusing on subjects from the common curriculum. The research will explore the impact of VR- and AR-based learning environments on student engagement, understanding, and motivation.

The study period spans from 2018 to 2025, during which the VR platform will be developed, tested, and evaluated in pre-selected educational institutions. The project will target high school students and teachers, analyzing their experiences with VR- and AR-mediated learning in real classroom contexts.

The prototype will provide interactive foundational learning modules, allowing students to explore concepts from various subjects. Additionally, the project will include a web application for educators, enabling the customization of VR and AR lessons, as well as the integration of learning analytics features.

The research will primarily focus on the pedagogical effectiveness of VR and AR in the teaching-learning process, without delving deeply into the logistical or economic aspects related to large-scale adoption of these technologies in educational systems. The feasibility of implementation will be analyzed based on selected schools, with no extension to non-academic applications of the technology.

2 BUSINESS ANALYSIS AND STRUCTURING

2.1 MARKET STUDY

Virtual reality and augmented reality hold significant potential to transform and enhance various activities across different contemporary contexts, even when used merely as support tools. Thus, an analysis of the current market can highlight the relevance of virtual reality in education, as well as outline future directions for its application. This section aims to address the questions raised in this study through market research and a literature review based on relevant authors and sources. The primary focus is to approach virtual reality as an educational innovation, analyzing its impact on students and teachers, as well as its contributions to content generation and information assimilation.

In this context, it becomes essential to analyze programs and technological applications aimed at educational use for both teachers and students, regardless of the subject in focus. The objective here is to identify and characterize possible applications of these technologies in the market and in society, establishing a perceptual overview of the state of virtual reality in education and its effectiveness. Furthermore, the reach of these solutions across different fields of knowledge will be examined, as well as their evolution as educational resources. The programs analyzed vary in terms of the sample's target group, the method of application, and the disciplines involved in the interventions.

Thus, the results of this research may serve as a basis for mapping the educational market regarding the adoption of virtual and augmented reality. Additionally, the study enables the identification of market strategies and applications aimed at implementing VR- and AR-based solutions in education, analyzing their social benefits and their competitiveness within the sector.

Table 1 - Market Research

Author/Year	Disciplinary Area	Sample	Students Characteristics	Teaching Protocol	XR Method	Main Results
CARVALHO, J. M./2020	Mathematics and Geography	50	1st-year public high school students; 15 to 17 years old	Mobile app + AR	LandscapAR	Increased group work and successful appropriation of geographic and mathematical concepts at a cognitive level
SOARES, Fredson; SANTANA, José; SANTOS, Maria/2022	Spatial Geometry	16	Teachers and undergraduate Pedagogy students at UFC	Software + AR + SF Methodology	GeoGebra	Greater engagement, increased support for pedagogical practice, more attractive and fun classes, and significant contribution to learning
CEN, Ling et al./2020	Organic molecules and inorganic reactions (Chemistry)	45	2nd-year high school chemistry students	Mobile app + AR	AIR-EDUTECH	Better content understanding and knowledge retention gains
GAN, Hong et al./2018	Redox reaction between hydrogen peroxide and sodium hypochlorite solutions (Chemistry)	10	2nd and 3rd-year high school students studying chemistry and biology for college entrance exams	Mobile app + AR	Unity e AR Vuforia kit (SDK)	Improved adaptation to different classroom contexts for teachers, increased student confidence in handling chemicals, and greater familiarity with lab environments

NEIVA, Tatiana/2023	Brazilian Biomes (Geography)	15	6th-grade elementary school students	Mobile app + VR	CardBoard Glasses + Bioma360	More attractive classes and fewer student difficulties in the subject, but teachers showed difficulty handling the technology
FURTADO, Priscila; NUNES, Renata/2021	Combustion reaction (Chemistry)	18	1st and 3rd-year high school students	Mobile app + VR	CardBoard + Google Expeditions	Significant improvement in content comprehension, more excitement and satisfaction during classes
TEIXEIRA, Nicole; CAMPOS, Aline/2019	Colors and pronunciation (English)	28	Preschool children (ages 5 to 6) in Pre-I and Pre-II	Mobile app + VR	BOBO VR Z4 + QuiverVision	Greater interaction between teacher and students, and enhanced learning, promoting higher student engagement
TARANILLA, Rafael et al./2019 /2019	Roman Civilization (History)	98	4th-grade elementary school students	Mobile app + VR	VirTimePlac	Statistically significant differences in favor of students who used Virtual Reality, both in motivation and academic performance
DEMITRIADOU, Eleni; STAVROULIA, Kalliopi; LANITIS, Andreas/2019	Geometric solids (Mathematics)	30	4th, 5th, and 6th-grade elementary school students; ages 9 to 11	Mobile app + VR + AR	ENTiTi Creator	Improved interactivity and student interest, contributing to more efficient learning and understanding of mathematical concepts

Source: Author's own work.

The mapping revealed largely positive results, highlighting the benefits of this technology in the learning process across various disciplines. However, challenges related to the implementation and accessibility of these resources were also identified. The detailed analysis of the collected data allows for a complete response to the first three research questions and provides an introductory insight into the last two central issues of this investigation.

The analyzed studies demonstrated that virtual and augmented reality are being applied in different areas of knowledge, including mathematics, geography, chemistry, history, and English. In all cases, there was an increase in student engagement, greater knowledge retention, and improved understanding of abstract concepts.

Nevertheless, some challenges were identified. Access to devices such as smartphones was not universal among students, which, in one case, led to the use of the teachers' and researcher’s personal devices to enable the educational experience. Additionally, the infrastructure of educational institutions was an obstacle in another case, with issues in configuring the school’s internet to support the applications in use. The school’s internet also showed instability at times, causing connection drops and slowdowns during the application download on students’ phones, which impacted the progress of the lesson.

Another relevant point was the difficulty reported by some teachers in learning and using these technologies. Although they acknowledged the educational benefits of VR and AR, the lack of proper training may increase lesson preparation time and hinder large-scale adoption. Furthermore, all studies used pre-built applications, without allowing teachers to customize the content, which may limit pedagogical flexibility.

The effectiveness of using VR and AR in education depends on the proper integration of these resources into pedagogical practice. The studies indicate that teachers can maximize the experience and effectiveness of the materials by using these technologies as interactive visual support, facilitating the understanding of abstract and practical concepts.

In this context, technology acts as essential support for specific subject content that requires a visual or practical approach for better understanding, always aligned with the lesson plan. The main factor enabling teachers to effectively use VR and AR to enhance their didactic materials lies in adaptation. Since the currently available technologies are not customized for each teacher, it becomes necessary for them to adjust their lessons to integrate these tools. However, this need for adaptation can become a discouraging factor, especially for teachers who already have a well-structured plan and do not find an adequate match between the existing technologies and their teaching methods.

It was observed that the gradual implementation of these technologies, distributed over multiple classes, facilitated the integration of the resources into the lesson plan. This progressive process allowed teachers to explore the possibilities of VR and AR in a more structured way, ensuring more efficient use and maximizing both the students' learning experience and the effectiveness of the didactic materials used.

It is also observed that the disciplines most prone to the implementation of virtual and augmented reality are those that demand a visual or practical approach for better content comprehension. Among these disciplines, the following stand out:

a) Mathematics: The use of VR is particularly effective in exploring geometric shapes and spatial concepts, as demonstrated by the study of Soares et al. (2022), which used the GeoGebra software to make lessons more dynamic and interactive.
b) Geography: Technologies such as LandscapeAR, studied by Carvalho (2020), facilitated the understanding of landforms, biomes, and geographical location, allowing students to visualize concepts in three dimensions.
c) History: Virtual reality can be widely used to recreate historical events and immersive experiences in cultural environments, as demonstrated by Taranilla et al. (2019) using technologies such as VirTimePlace.
d) Chemistry: The discipline benefits from augmented and virtual reality by enabling the interactive visualization of molecules and chemical reactions, as demonstrated in the studies by Cen et al. (2022) and Gan et al. (2018).
e) Physics: Although not addressed in the analyzed studies, VR can facilitate the understanding of physical laws by enabling the simulation of laboratory experiments and natural phenomena in a controlled environment.

Although the analyzed studies did not extensively address the acceptance of students and teachers, the data suggest a generally positive reception. Teachers recognize the positive impact of these technologies on education and tend to accept their implementation, despite initial learning difficulties. Students, being digital natives, show ease and enthusiasm in using these tools, which contributes to a more dynamic learning environment.

Based on the Theory of Reasoned Action, Davis (1989) proposed the Technology Acceptance Model (TAM) with the aim of identifying the determining factors for individuals' adoption of new technologies. According to Davis, attitudes toward a specific technology are key elements in the decision to use it.

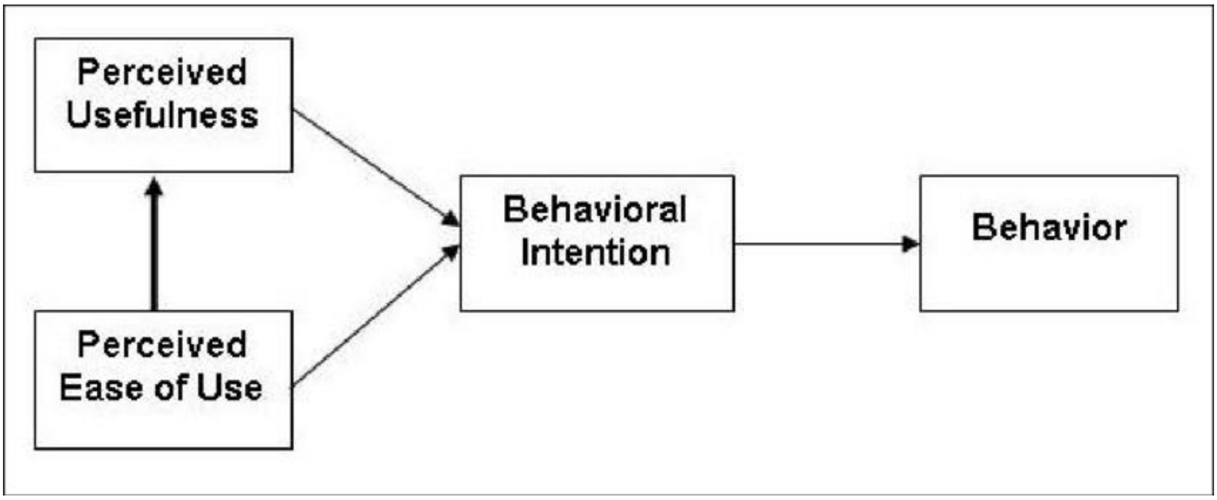

Figure 1 Technology Acceptance Model (TAM)
Source: Davis-1989.

In the TAM model, the author establishes two main factors that influence the acceptance and use of a technology: (1) perceived usefulness, which refers to the degree to which an individual believes that a given technology can improve their performance in activities; and (2) perceived ease of use, a latent construct that directly impacts an individual's willingness to adopt the technology.

The literature review conducted supports this model, highlighting a broadly positive reception toward educational technologies. This acceptance can be explained by the perception of usefulness, as both teachers and students recognize the beneficial impact of these tools on the teaching and learning process. Furthermore, even in the face of initial challenges related to adaptation and learning how to use these technologies, there is a predisposition toward their adoption, which aligns with the concept of perceived ease of use in the TAM model.

According to Prensky (2001), the current generation of digital natives is predominantly composed of individuals who exhibit familiarity and confidence when facing challenges posed by Information and Communication Technologies (ICT), actively exploring the multiple possibilities provided by technological advancements. In this context, Generation Y's inherent interest in experimentation and discovery should be leveraged in the educational environment to guide teaching and learning toward an approach that fosters effective dialogue with new technological tools, promoting greater interaction between students and the digital resources available.

The studies did not directly focus on the resources required to maintain these platforms, but several challenges were noted. In one of the works reviewed, the teacher and researcher provided their own mobile phones so that students could access the applications, highlighting the lack of devices available for all students involved in the class. Issues were also reported with the school's internet, which delayed the lesson due to the need for specific configurations to run the applications. Additionally, the learning curve faced by teachers in using new technologies was an obstacle, reinforcing the need for proper training.

The universalization of access to Information and Communication Technologies remains one of the central challenges in Brazilian education, especially concerning the inclusion of all students, teachers, and educational institutions in this process. According to Kenski

(2012), the effective integration of ICT into the curriculum must go beyond the mere introduction of new tools, requiring a pedagogical overhaul that transforms schools and classrooms into dynamic spaces for learning, experimentation, and civic education. However, this reality is still distant for many schools, particularly in the public system, which face difficulties even in implementing basic technologies such as stable internet access, communication infrastructure, and the availability of computers for pedagogical use.

In this context, the adoption of more advanced technologies, such as Virtual Reality and Augmented Reality headsets, presents even more complex challenges. The implementation of these resources not only requires financial investment in the acquisition of devices but also demands teacher training, ongoing technical support, and methodological adaptation to ensure their effective pedagogical use. Furthermore, as pointed out by Valente (2019), the introduction of new educational technologies must be accompanied by public policies that ensure equitable access; otherwise, there is a risk of deepening educational inequalities between schools with greater and lesser infrastructure. Thus, in institutions that do not even have the basic conditions for digitalizing education, the implementation of VR and AR may be unfeasible or result in limited and disconnected use from traditional pedagogical practices.

A The literature review highlighted that VR and AR offer significant benefits for education, especially in subjects that require a visual and experiential approach, such as mathematics, geography, history, and chemistry. These technologies expand learning possibilities by making content more interactive and immersive, thereby enhancing student comprehension and engagement. However, their implementation still faces considerable challenges, particularly in institutions lacking adequate technological infrastructure.

Financial, technical, and human resource limitations are especially evident in public schools, where basic resources such as stable internet access, computers, and technical support for teachers are often lacking. Furthermore, the adoption of these technologies demands significant effort from educators, who must adapt their lesson plans to the available tools, as the applications reviewed in the literature do not allow for content customization. Although teachers recognize the potential of VR and AR for education, difficulties in learning and adapting to these tools may pose barriers to their effective implementation.

Another relevant aspect is the reliance on mobile devices for the use of these technologies, given that not all students have access to smartphones suitable for this purpose. Additionally, issues such as unstable school internet and technical difficulties in configuring the applications can hinder the learning experience, delay the progress of lessons, and limit the effectiveness of digital resources.

Considering these challenges, the implementation of VR and AR in education should be carried out in a planned and gradual manner, ensuring that teachers receive adequate training and that educational institutions are equipped with the necessary infrastructure to support the use of these technologies. The conscious and structured adoption of these tools can significantly contribute to a more dynamic, interactive, and accessible learning environment, expanding educational possibilities and promoting a more engaging experience for students.

2.2 PROPOSED SOLUTION

This chapter presents the formulation of an innovative application aimed at addressing the challenges related to student motivation and engagement in the teaching-learning process. The proposal results from an in-depth market analysis and a critical review of existing solutions in the literature, aiming to integrate already-tested elements with new functionalities and hypotheses developed by the author.

As evidenced in the market research section, the literature presents various approaches to the problem of motivation and engagement, as well as to the improvement of knowledge comprehension and retention. These solutions vary according to the institution's profile, user characteristics, and the subjects involved. Thus, the proposal presented here is the result of the combination of successful experiences, the overcoming of identified difficulties, and the incorporation of innovations aimed at enhancing the pedagogical process.

2.2.1 PROBLEM DEFINITION

The proposed solution, entitled interactive virtual environment for secondary schools: application of virtual and augmented reality in the teaching-learning process, aims primarily to provide teachers with an intuitive tool for the creation and availability of immersive pedagogical content. Through this platform, educators will be able to develop virtual reality and augmented reality experiences that support teaching materials and lesson plans, centralizing student access.

The project encompasses four user groups or personas: a) teacher: the central agent, responsible for creating and adapting supporting content; b) student: the beneficiary of the improved learning experience, albeit with a passive role; c) educational institution: user and promoter of the tool, integrating it into its pedagogical structure; d) content producers: teachers or external professionals who can contribute to the creation and availability of immersive experiences.

Among the difficulties and needs identified through market research, the following stand out:

a) Low Levels of Motivation and Engagement: Student demotivation, which undermines the teaching-learning process.
b) Limitations in Visual Subjects: Subjects that demand specific visual resources and practical activities are hindered by traditional methodologies.
c) Demand for Active Methodologies: It is essential to adopt more dynamic pedagogical approaches capable of meeting the needs of a generation immersed in digital technologies.
d) Rigidity in VR/AR Experiences: Current solutions impose the adaptation of pedagogical planning to the technological experience, rather than allowing flexible resources to adapt to the classroom context.
e) Insufficient School Infrastructure: Many institutions lack the technological and connectivity resources necessary to support online VR/AR solutions.

f) Technical Training for Teachers: The lack of time and adequate training for the use of immersive technologies represents a significant barrier.

2.2.2 SOLUTION'S MAIN FEATURES

The proposed platform integrates a series of features aimed at facilitating and optimizing the process of creating, managing, and sharing immersive educational content through the following functionalities:

a) Creation of Immersive Content: Enables teachers to develop didactic support materials in virtual and augmented reality without requiring in-depth technical knowledge.
b) Multiplatform Accessibility: Content will be made available in an adaptive format, allowing usage through virtual reality headsets in school environments or via mobile devices when accessed in other contexts.
c) Integration and Sharing: Allows both teachers and other content producers to publish their creations on the platform, classifying them as either free or paid.
d) Request for Customized Content: Offers the option for institutions or educators, who do not wish to produce their own material, to submit specific requests for the creation of immersive experiences.
e) Integrated Class Management: Enables the creation of virtual classes, facilitating the organization and access to content according to permissions defined by the institution.
f) Online Synchronization and Availability: Ensures that content is managed exclusively through the online environment, with periodic synchronizations to VR devices, guaranteeing data integrity.
g) Pre-developed Basic Content: Provides foundational materials for core subjects (History, Geography, Chemistry, Physics, and Mathematics) to support initial implementation and pedagogical use.

2.2.3 PROJECT TIME MANAGEMENT AND PLANNING

The objective of project time management planning is to develop and control the project schedule. This process will occur at two distinct stages: the first takes place at the beginning of the project, when technical knowledge is still limited. At this stage, initial estimates will be produced—even if broader in scope—to support the planning of activities. The second stage occurs later, once specific technical information has been obtained, allowing for the refinement of the previously generated estimates.

In the initial estimation phase, only the macro activities related to the development of the module will be considered. During the refinement phase, sub-activities that were not previously identified—or that may emerge throughout the development process—will also be included.

This entire process comprises five key activities: (I) Identifying project dependencies; (II) Estimating the duration of project activities using a top-down approach; (III) Developing the project schedule; and (IV) Controlling the project schedule.

The logical relationships, interactions, and interdependencies between activities were defined in the macro-division of the four modules. This division reflects the fact that the successful execution of a given module depends entirely on the successful completion of the preceding module, thus establishing a temporal dependency among them.

Table 2 - Project Time Planning

1st Module	**Focal Point:** Proposal and Solution Planning
Description ➞	Stage dedicated to defining the problem, justification, objectives, and requirements gathering, focusing on project proposal and planning.
Precondition ➞	Definition of the project partner.
Week 1 (03/Feb/2025 to 07/Feb/2025) ➞	Project Charter and Choice of Follow-up Instructions
Week 2 (10/Feb/2025 to 14/Feb/2025) ➞	Introductory Topics: Introduction, Problem Statement, and Justification
Week 3 (17/Feb/2025 to 21/Feb/2025) ➞	General and Specific Objectives; Premises and Restrictions
Week 4 (24/Feb/2025 to 28/Mar/2025) ➞	Delimitation of the Objective and Market Research – Literature Review
Week 5 (03/Mar/2025 to 07/Mar/2025) ➞	Consolidation of Results Addressing the Problem Statement
Week 6 (10/Mar/2025 to 14/Mar/2025) ➞	Project Proposal and Specification of Main Features
Week 7 (17/Mar/2025 to 21/Mar/2025) ➞	Project Time and Cost Planning
Week 8 (24/Mar/2025 to 28/Mar/2025) ➞	Business Model Canvas and Empathy Map
Week 9 (31/Mar/2025 to 04/Apr/2025) ➞	User Cases
Week 10 (07/Apr/2025 to 11/Apr/2025) ➞	Functional and Non-functional Requirements and Final Document Review
2nd Module	**Focal Point:** Platform Development
Description ➞	Focused on platform construction, defining architecture, reference testing, feature development, and prototype creation.
Precondition ➞	Solution proposal finalized.
Week 1 (21/Apr/2025 to 25/Apr/2025) ➞	Testing Applications Found on Quest and Creating a Navigable Prototype
Week 2 (28/Apr/2025 to 02/May/2025) ➞	Architecture and Technical Specification of the Solution
Week 3 (05/May/2025 to 09/May/2025) ➞	Backend Development: Main APIs and Authentication
Week 4 (12/May/2025 to 16/May/2025) ➞	Initial Frontend Structure Development
Week 5 (19/May/2025 to 23/May/2025) ➞	Frontend and Backend Integration; User and Permission Structuring
Week 6 (26/May/2025 to 30/May/2025) ➞	Implementation of Immersive Content Creation System
Week 7 (02/Jun/2025 to 06/Jun/2025) ➞	Development of Class and Lesson Management System
Week 8 (09/Jun/2025 to 13/Jun/2025) ➞	Online Synchronization with VR Devices and Connectivity Testing
Week 9 (16/Jun/2025 to 20/Jun/2025) ➞	Frontend and Backend Integration Testing; Improvements Based on Results
Week 10 (23/Apr/2025 to 26/Jun/2025) ➞	Technical Documentation and Beta Version Delivery
3rd Module	**Focal Point:** Market Fit and Project Adjustments
Description ➞	Validation stage with users, identifying improvements and making platform adjustments to better align with market needs.
Precondition ➞	Functional prototype developed.
Week 1 (04/Aug/2025 to 08/Aug/2025) ➞	Feedback Collection from Teachers and Advisor
Week 2 (11/Aug/2025 to 15/Aug/2025) ➞	Feedback Analysis and Improvement Prioritization
Week 3 (18/Aug/2025 to 22/Aug/2025) ➞	UX Adjustments and Performance Improvements

Week 4 (25/Aug/2025 to 29/Aug/2025) ➞	Implementation of Newly Identified Requirements
Week 5 (01/Sep/2025 to 05/Sep/2025) ➞	Addition of Core Subject Content
Week 6 (08/Sep/2025 to 12/Sep/2025) ➞	Application Testing with Teachers
Week 7 (15/Sep/2025 to 19/Sep/2025) ➞	Monitoring and Data Collection on Usability
Week 8 (22/Sep/2025 to 26/Sep/2025) ➞	Final Corrections Based on Real Tests
Week 9 (29/Sep/2025 to 03/Oct/2025) ➞	Preparation for Final Version Release
Week 9 (06/Oct/2025 to 09/Oct/2025) ➞	Final Documentation Review and Module Closure
4th Module	**Focal Point:** Final Project Validation
Description ➞	Project conclusion with final testing, full solution validation, delivery preparation, and documentation of the developed product.
Precondition ➞	Platform adjusted and ready for formal validation.
Week 1 (14/Oct/2025 to 17/Oct/2025) ➞	Final Evaluation Planning: Criteria, Methods, and Metrics
Week 2 (20/Oct/2025 to 24/Oct/2025) ➞	Schedule Deployment in a Real School Environment (Subject to Availability)
Week 3 (27/Oct/2025 to 31/Oct/2025) ➞	Analysis of Obtained Results
Week 4 (03/Nov/2025 to 07/Nov/2025) ➞	Final Platform Version Development Based on Evaluation (if necessary)
Week 5 (10/Nov/2025 to 14/Nov/2025) ➞	Final Validation with Advisor and Partner
Week 6 (17/Nov/2025 to 21/Nov/2025) ➞	Final Presentation Preparation (Pitch + Demo)
Week 7 (24/Nov/2025 to 28/Nov/2025) ➞	Final Report Review and Documentation Validation
Week 8 (01/Dec/2025 to 05/Dec/2025) ➞	Technical and Scientific Project Delivery
Week 9 (08/Dec/2025 to 12/Dec/2025) ➞	Buffer Week for Final Adjustments and Training
Week 10 (15/Dec/2025 to 19/Dec/2025) ➞	Official Presentation and Project Closure

Source: Author's own work.

2.2.4 PROJECT COST MANAGEMENT AND PLANNING

The objective of project cost management is to develop and control the project’s budget. Similar to time management, this process will be carried out in two distinct stages: initially, with a preliminary identification of potential sources of expenditure—albeit without specific estimates—and subsequently, in a more detailed manner, once there is greater clarity regarding time, effort, and resource allocation estimates.

This process will be divided into three stages: (I) Identifying potential sources of project expenses; (II) Developing the project budget; and, (III) Controlling the project budget.

Preliminary cost estimates will be categorized into hardware resources, software resources, human resources, and other expenses. In this project, three potential cost sources have been identified: the first refers to the acquisition of specific assets not available for free, given that the project's focus is not on the full authorial development of 3D assets; the second involves the implementation of the project in cloud-based systems and architectures; and the third concerns resources required for conducting tests in real-world environments, such as transportation or the purchase of support materials for classroom testing. However, due to the early stage of the project, it is not yet possible to provide accurate estimates for any of these expenses.

Throughout the project’s development, analyses and comparisons will be conducted whenever necessary, including documentation of budgetary variations or the emergence of new expenditures—whether positive or negative. The goal is to mitigate risks and ensure that no cost poses a threat to the viability of the project.

Throughout the development of the project, the schedule will be continuously compared, analyzed, and revised whenever necessary. Any deviations will be recorded, and the schedule will be updated accordingly—either by incorporating recovery actions or by adapting it to a new version. From that point on, new decisions will be made after careful consideration of the potential impacts of these changes on the timeline and on the stakeholders involved in the project.

2.3 LEAN CANVAS

The Business Model Canvas (BMC), proposed by Osterwalder and Pigneur (2011), represents an innovation by allowing entrepreneurs to visualize, in an integrated manner, the main strategic aspects of their business in a single framework. This model organizes the essential elements of a company into nine blocks: customer segments, value proposition, distribution channels, customer relationships, revenue streams, key resources, key activities, strategic partnerships, and cost structure (OSTERWALDER; PIGNEUR, 2011).

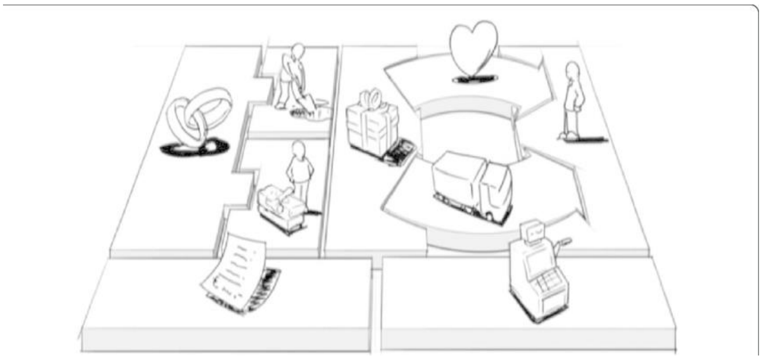

Figure 2 Key Components of the Business Model Canvas
Source: Osterwalder e Pigneus (2011)

One of the main advantages of the Business Model Canvas is its simplicity and objectivity. By consolidating these nine elements into a single visual scheme, it provides a comprehensive view of the project, facilitating the identification of potential gaps, opportunities for improvement, and strengths that can be further developed. In this way, its application in this project enables the analysis of the main strengths, weaknesses, opportunities, and threats associated with its development, contributing to more strategic and evidence-based management.

a) Customer Segments: Defines the different groups of individuals or organizations that our project aims to reach and serve.
b) Value Proposition: Describes the set of products and services offered by us, which generate value for the specific customer segments.
c) Channels: Represent the means through which our project intends to interact with our customers. These include communication, sales, and distribution channels, serving as the main interface between us and our target audience.

d) Customer Relationships: Establishes the types of relationships we intend to maintain with the different customer segments.
e) Revenue Streams: Describes the ways in which the project can generate profit from each customer segment.
f) Key Resources: Represent the main assets required for the Business Model to function properly.
g) Key Activities: Details the fundamental actions for delivering the Value Proposition, as well as those essential to the operation of the project.
h) Key Partnerships: Identifies the main partners and suppliers of the project, specifying the resources acquired through these partnerships and the activities performed by the partners that are essential to the operation of the business.
i) Cost Structure: Describes all the costs involved in the implementation and maintenance of the Business Model.

Table 3 - Lean Canvas

<table>
<tr><th>Key Partnerships</th><th>Key Activities</th><th>Value Proposition</th><th>Customer Relationships</th><th>Customer Segments</th></tr>
<tr>
<td rowspan="3">Public and private educational institutions.

EdTech and augmented/virtual reality companies.

Universities and research centers.

Departments of education and government agencies.
Independent educational content creators.

Cloud storage and infrastructure platforms.</td>
<td>Platform development and maintenance.

Creation and curation of immersive content.

Technical and pedagogical support.

Management of partnerships with schools and content creators.

User training and capacity building.

Monitoring and evaluation of usage for continuous improvement.</td>
<td rowspan="3">Intuitive platform for creating and sharing immersive content (VR/AR) without the need for advanced technical knowledge.

Multi-platform accessibility, allowing use with VR headsets or smartphones.

Customizable experiences, tailored to institutional realities and pedagogical needs.

Pre-developed content for core subjects, making initial adoption easier.

Option for on-demand content creation for institutions and teachers.

Centralized environment for managing classes and lessons with online synchronization.</td>
<td>Online technical and pedagogical support.

Collaborative community among teachers and content creators.

Tutorials and training for teacher development.

Personalized service for institutions with on-demand plans.

Continuous feedback for platform improvement.</td>
<td rowspan="3">High school teachers looking to create more immersive and engaging classes.

High school students who benefit from active and visual learning methods.

Educational institutions interested in modernizing their teaching methodology.

Educational content creators, such as expert teachers and education professionals who want to monetize their materials.</td>
</tr>
<tr><th>Key Resources</th><th>Channels</th></tr>
<tr>
<td>Technological platform (back-end, front-end, cloud servers).

Software development team.
Pedagogical team and instructional design specialists.

Content library and 3D assets.

Infrastructure for testing and synchronization with VR/AR devices.</td>
<td>Official platform website.

Mobile app (Android/iOS).

Partnerships with departments of education and schools.

Educational events, tech fairs, and pedagogical conferences.

Social media and content marketing.</td>
</tr>
</table>

<table>
<tr><th>Cost Structure</th><th>Revenue Streams</th></tr>
<tr>
<td>Cloud infrastructure and server costs.
Acquisition/licensing of 3D assets.
Salaries and hiring of technical and pedagogical teams.
Platform development and testing.
Training and technical support costs.
Materials for school testing (VR headsets, mobile devices).
Marketing and communication expenses.</td>
<td>Monthly/annual subscriptions for educational institutions.
Sale of premium content by partner creators.
On-demand production services for immersive experiences.
Public and private partnerships focused on educational innovation.
Licensing for school networks or governments.</td>
</tr>
</table>

Source: Author's own work.

3 CONCEPTUAL DESIGN OF THE SOLUTION

3.1 EMPATHY MAP

The Empathy Map is a design thinking tool used mainly in the early stages of the solution development process. Its aim is to identify the characteristics of the target audience, also known as the persona. Thus, this section of the project aims to use the Empathy Map tool to outline the profile of the main individuals involved in the proposed solution, which focuses on the use of virtual and augmented reality as a pedagogical resource. Figure 3 shows the Empathy Map, made up of six areas that guide the analysis of the feelings and perceptions of those involved in the solution, based on the following questions in each quadrant:

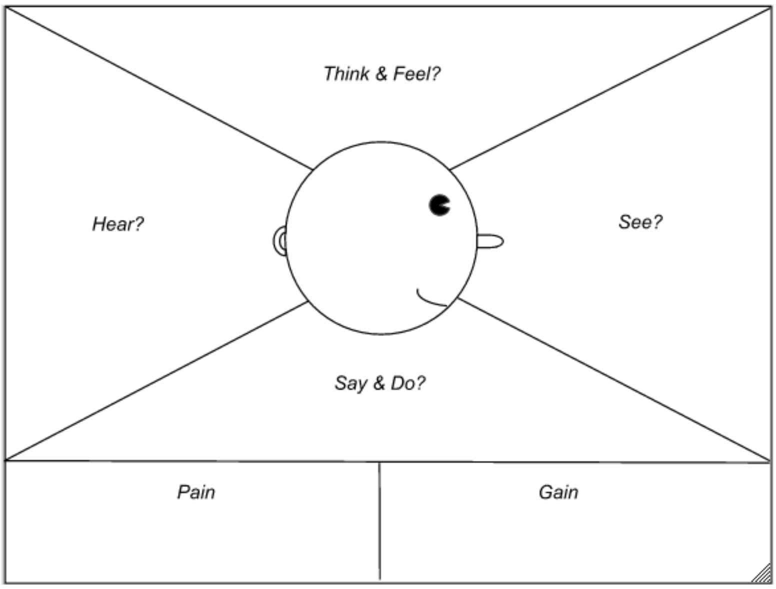

Figure 3 Empathy Map
Source: Dave Gray (2017)

The empathy map is structured into six fundamental areas that allow us to understand the perceptions, feelings and behaviors of a target audience in depth. The first area concerns what the individual listens to, including the opinions and advice of friends, family and work colleagues, as well as the messages transmitted by their superiors and influencers. It also encompasses the communication products and services they regularly consume. The second area covers what you think and feel, analyzing the ideas aroused by a product or service, your feelings about the world, your main concerns and the dreams you have. The third area investigates what you see, considering the environment in which you live, the visual elements of your daily life, the characteristics of the people around you and the options offered by the market related to the problem in question. In addition, this area explores the visual stimuli you perceive and how they impact your decisions.

The fourth area of the empathy map examines what the individual says and does, evaluating their attitudes in public, their appearance and daily behavior, the way they relate to others and the speeches they usually adopt. It also analyzes their practices, actions and occupations. The fifth area deals with pain, identifying the main fears, frustrations and obstacles that need to be overcome in order to achieve your goals. Finally, the sixth area investigates the gains, exploring your desires, daily needs, how you measure success, the challenges to be faced and the elements that could eliminate your problems.

The Empathy Map was specifically developed for three personas: teachers, students, and the school. This decision was guided by insights gathered during the market research, which clearly highlighted the key pain points and needs of teachers. While students and schools also face relevant challenges, their experience is indirectly affected — serving as a positive side effect of the proposed solution.

Since the platform is primarily designed to support and enhance the creation of immersive educational materials by teachers, the Empathy Map focuses specifically on this user group. The resulting profiles of each mapped persona are presented below.

Table 4 - Empathy Map

Teacher	
What do they think and feel?	The teacher understands that technologies like Virtual Reality and Augmented Reality have the power to transform education by making lessons more engaging and effective. They see these tools as particularly valuable for teaching abstract or highly visual content. However, they often feel discouraged when forced to adapt lessons to tools that don't fit their educational context, and they struggle to integrate these technologies into already established teaching methods.
What do they see?	In their daily experience, the teacher sees various technical barriers: poor infrastructure, unstable internet, and a shortage of available devices. They notice that VR and AR applications lack personalization and that many students have unequal access to smartphones or other equipment. They also observe that fellow teachers face the same implementation challenges.
What do they say and do?	The teacher shows interest in using VR and AR when they identify clear benefits to learning. They speak openly about their difficulties, such as the lack of training, and sometimes use their own devices to provide immersive experiences in the classroom. Their approach involves introducing the technologies gradually, testing and refining their use across multiple lessons.
What are their pains?	The teacher's main pains include insufficient training on how to use VR and AR, increased lesson planning time due to the need for adaptation, and limited access to devices among students. Unstable school internet is another major obstacle, as is the challenge of adapting content without customizable tools.
What are the benefits?	The teacher hopes to use technology to make lessons more engaging, help students grasp complex concepts, and improve knowledge retention. They aim to incorporate VR and AR as supportive, interactive visual tools aligned with the lesson plan, creating a more dynamic and inclusive learning environment.
What do they hear?	They hear students expressing excitement about using new technologies, which highlights the engagement potential of these tools. They also hear institutional narratives promoting educational innovation and digital transformation, although practical support for implementing these changes is often lacking, both technically and pedagogically.
Student	
What do they think and feel?	The student shows enthusiasm when interacting with technologies like VR and AR, especially since they are already familiar with digital devices. They feel more engaged and motivated during lessons that use these tools. Experiencing content in a visual and immersive way facilitates understanding, particularly of abstract concepts. However, when there are technical failures or when they don't have the necessary equipment, they may feel frustrated and excluded.
What do they see?	The student sees technology as an ally in learning, making classes more interesting and helping them understand complex content. They notice that in some situations, there are limitations, such as having to use the teacher's phone or dealing with school internet issues. This gap between the ideal and the reality can generate a sense of inequality in access to digital learning.
What do they say and do?	During immersive technology lessons, the student interacts with more autonomy and curiosity, actively participating in activities. They talk with peers about how the classes have become more fun and how they better understood the content. On the other hand, if they don't have access to the device or face issues, they express discouragement and frustration.
What are their pains?	The student's pains are related to the lack of access to compatible devices, unstable internet connection, and frustration caused by interrupted technology-based activities. They may also feel left behind when unable to fully participate due to technical limitations.
What are the benefits?	The main gains for the student include increased engagement, easier understanding of complex topics, greater motivation to participate in lessons, and the feeling that learning is

	more aligned with their digital reality. The use of VR and AR contributes to a more engaging and meaningful educational experience.
What do they hear?	The student hears teachers' instructions on how to use the applications, as well as positive comments from classmates about the experience with VR and AR in class. At the same time, they hear complaints about technical issues and difficulties using the tools, which makes them aware of the school's structural instability.
School	
What do they think and feel?	The school recognizes the pedagogical potential of Virtual and Augmented Reality technologies, understanding that their adoption could represent an advancement in the teaching-learning process. However, it feels limited by structural difficulties and the lack of financial, human, and technical resources needed for the effective implementation of these technologies.
What do they see?	The school finds itself in a scenario where many of its units lack even the basics for digital education, such as stable internet access or enough computers. It also sees the inequality between public and private schools, with the former facing greater obstacles in incorporating innovative technologies.
What do they say and do?	Despite the limitations, the school seeks strategies to gradually implement technologies. In some cases, it relies on the efforts of teachers and researchers to make VR and AR experiences feasible. It tries to adapt to the demands of the contemporary educational landscape, even with limited resources, and emphasizes the importance of public policies to ensure digital inclusion.
What are their pains?	Institutional pains include the lack of adequate infrastructure, insufficient public investment, shortage of mobile and technological devices, unstable internet, and the lack of ongoing teacher training. Added to this is the challenge of maintaining the use of technology in a sustainable and curriculum-integrated way.
What are the benefits?	Despite the challenges, the school identifies that the use of VR and AR can enhance teaching quality, promote greater student engagement, and make the school environment more innovative. By investing in teacher training and infrastructure, it can offer a more inclusive, modern, and effective educational experience.
What do they hear?	The school hears educational guidelines encouraging the inclusion of digital technologies, reports of best practices, and pressure for innovation in teaching. At the same time, it hears about the difficulties faced by teachers, such as lack of training and increased workload, and by students, especially regarding the lack of devices and unstable connections.

Source: Author's own work.

3.2 USER CASES

A use case can be defined as a narrative description of a sequence of events that occurs when an actor — understood as an external agent to the system — interacts with the system in order to perform a specific task (JACOBSON, BOOCH, and RUMBAUGH, 1999). More precisely, it represents a structured sequence of steps or operations carried out during the interaction between the actor and the system, aiming to achieve a goal established by the actor. In this context, the actor refers to any external entity that communicates with the system.

The primary purpose of use cases is to describe the system's functional requirements — that is, what the system is expected to perform — even though such descriptions may not encompass the entirety of the system's requirements.

In addition to providing a more detailed overview of the system's behavior, use cases facilitate a deeper understanding of its functionalities and interactions, thereby contributing significantly to the analysis of the project's complexity. Furthermore, they serve as an effective tool for addressing complex problems, as they allow for the clear visualization of potential solutions.

Throughout the development process, use cases also support progress tracking, enabling the identification of functionalities that are operating as intended and those requiring adjustments. This continuous analysis enhances the quality of the project by enabling the correction of issues and the implementation of improvements in accordance with user needs.

Table 5 - User Cases

ID: UC001	**User Case Name:** Usability and Intuitive Interface
Description ➞	The purpose of this use case is to allow the teacher to create virtual or augmented reality content on the platform
Primary Actor:	Teacher
Precondition:	Teacher authenticated in the system
Main Scenario ➞	1. The teacher accesses the creation area 2. The system displays the creation options 3. the teacher enters the necessary data and media 4. The teacher saves and publishes the content
Post-condition:	Immersive content available on the platform
Alternative Scenario:	b- Content upload failed
Inclusion:	UC002 - Authenticate User
Extension:	UC003 - Publish Content
ID: UC002	**User Case Name:** Authenticate User
Description ➞	Validate the user in the system (teacher, student or content producer)
Primary Actor:	Actor User (Any persona)
Precondition:	None
Main Scenario ➞	1. The system requests email and password. 2. The system validates the credentials. 3. The system identifies the type of user. 4. The system directs the user to their dashboard.
Post-condition:	User authenticated and redirected
Alternative Scenario:	b- Invalid credentials.
Inclusion:	None
Extension:	UC005 - Recover Password
ID: UC003	**User Case Name:** Publish Content
Description ➞	Allow the content created to be published (free or paid).
Primary Actor:	Actor Teacher / Content Producer
Precondition:	Content created and saved.
Main Scenario ➞	1. The user selects the content. 2. The system offers publishing options (free or paid). 3. User confirms publication. 4. The system makes the content available on the platform.
Post-condition:	Content visible to students/authorized.
Alternative Scenario:	b- Communication failure with the server.
Inclusion:	UC001 - Create Immersive Content
Extension:	UC006 - Request Approval (if paid)
ID: UC004	**User Case Name:** Request Personalized Content
Description ➞	Allow schools or teachers to request personalized content.
Primary Actor:	Institution / Teacher
Precondition:	Authenticated user
Main Scenario ➞	1. The user accesses the requests area. 2. Fills in the data and the type of content desired. 3. sends the request to the system. 4. the system notifies the content producers.
Post-condition:	Request stored and awaiting a response.
Alternative Scenario:	b- Failure to send request.
Inclusion:	UC002 - Authenticate User
Extension:	UC003 - Publish Content

ID: UC005	**User Case Name:** Access Immersive Content
Description ➞	Allow students to access virtual or augmented reality content via the platform.
Primary Actor:	Student
Precondition:	Student authenticated and granted access to the content.
Main Scenario ➞	1. The student enters the platform. 2. The system displays the content available to them (according to class or permissions). 3. The student selects a piece of content. 4. The content is loaded and started (via Web, VR or AR).
Post-condition:	The student views or interacts with the immersive content
Alternative Scenario:	b- Student tries to access paid content without having purchased it c- Content loading error
Inclusion:	UC002 - Authenticate User
Extension:	None
ID: UC005	**User Case Name:** Approve Paid Content
Description ➞	Allow the requester (teacher or institution) to approve or reject paid content published by a producer following a request.
Primary Actor:	Requester (Teacher or Institution)
Precondition:	Paid content has been made available by a producer in response to a personalized request.
Main Scenario ➞	1. The requester accesses their requests and paid publications area. 2. The system displays the delivered content awaiting approval. 3. The requester reviews the material received. 4. The requester approves or rejects the content. 5. The system records the decision and notifies the producer. If approved, payment is processed and the content released.
Post-condition:	The content has been approved and released on the platform, or it has been rejected and returned to the producer with justification.
Alternative Scenario:	a. The requester asks for changes before approval. b. The content is rejected with justification. c. The requester does not respond, and the content remains in “awaiting approval”.
Inclusion:	UC003 - Publish Immersive Content UC004 - Request Custom Content Production
Extension:	None

Source: Author's own work.

3.3 FUNCTIONAL REQUIREMENTS

Functional requirements are a fundamental part of the project elicitation process. They describe the functions the system must perform, such as processing data, responding to inputs, performing calculations, and interacting with other systems or users. These requirements are essential for both the system design and effective communication between developers and users (PRESSMAN, 2010, p.63). Initially, all expected functions and actions will be defined, ensuring that the system meets the previously identified needs of the target audience. Each requirement is detailed with its name, description, category, priority, relevant information, and business rules.

Description of the categories:

a) Evident: The clear and expected behavior of the system.
b) Hidden: Functionality that is not directly perceptible to the user.
c) Legal: Requirements imposed by current laws or regulations.
d) Desirable: Items that, while not mandatory, add value to the overall experience.

Table 6 - Functional Requirements

ID: FR001	**Requirement Name:** Create Immersive Content
Description ➞	Allowing teachers to develop teaching materials in virtual and augmented reality without in-depth technical knowledge
Category: Evident	**Priority:** Essential
Information ➞	• Visual editor for VR/AR experiences with an intuitive interface • Library with pre-loaded 3D models and objects • Option to import external media (audio, video, images, animations) • Fields for adding pedagogical descriptions and learning objectives • Save and preview button
Business rules:	• Only users with a teacher or producer profile can create content • Saved content must be validated for compatibility with mobile devices and VR
ID: FR002	**Requirement Name:** Access Immersive Content
Description ➞	Allow students to access the content created on different devices
Category: Evident	**Priority:** Essential
Information ➞	• Adaptive interface for mobile devices and VR goggles • Listing of content by subject and school level • Authentication system for access control • Immersive viewer with simple commands (start, pause, exit)
Business rules:	• Only students linked to the class have access to the content released by an institution • The content accessed must be synchronized with the latest version available in the cloud
ID: NFR003	**Requirement Name:** Share Content on the Platform
Description ➞	Allow teachers and producers to publish their content on the platform, categorizing it as free or paid
Category: Evident	**Priority:** Essential
Information ➞	• Field for selecting the type of publication: free or paid • Submission form with title, description, subject, target audience • Administration interface for viewing and approving published content
Business rules:	• All published content must undergo technical and pedagogical compliance validation. • Paid content must follow the platform's policies and be linked to an account with a valid CNPJ (in the case of external producers) • Teachers linked to an institution can only share content with the school's internal audiences • The intellectual property of content produced by teachers linked to institutions for classes at the institution belongs to the institution, and cannot be made publicly available outside the institutional environment without the school's approval
ID: FR004	**Requirement Name:** Share Content on the Platform
Description ➞	Allow schools or teachers to request the development of personalized content
Category: Evident	**Priority:** Important
Information ➞	• Request form with details of the institution, educational objective, desired topic, deadlines and available budget • Notification system for accredited producers • Dashboard to track the status of the request (under analysis, accepted, in production, delivered)
Business rules:	• Only users with an institutional or teacher profile can request personalized content • The request must be approved by the platform administration before being sent to the producers
ID: FR005	**Requirement Name:** Manage Virtual Classes
Description ➞	Allow teachers to create and manage classes, controlling access to content by student
Category: Evident	**Priority:** Essential
Information ➞	• Register classes with name, subject, school year and institution • Linking students by access code or spreadsheet import • Individual permissions to access immersive content
Business rules:	• Only teachers linked to an institution and with the appropriate permissions can create and manage classes • Each class must be linked to at least one teacher

	• Non-students from the institution cannot access school content without explicit permission
ID: FR006	**Requirement Name:** Synchronize Online Content with Devices
Description ➞	Ensure that content is always available in an up-to-date form on users' devices
Category: Evident and hidden	**Priority:** Important
Information ➞	• Mechanism for periodic automatic synchronization with the cloud • Notifications of pending updates or synchronization failures
Business rules:	• All synchronization must guarantee data integrity
ID: FR006	**Requirement Name:** Synchronize Online Content with Devices
Description ➞	When creating didactic content linked to a class, lesson or pedagogical plan for an institution, teachers should be aware that this content is automatically considered the intellectual property of the educational institution. Therefore, this content cannot be made available to users outside the institution, not even by the creator, but only by the institution
Category: Legal	**Priority:** Important
Information ➞	• Content created in this context cannot be marked as public or made available outside institutional boundaries • Teachers will not be able to export, clone or share this content outside the institution via the platform • The institution may, at its own discretion, publish or share this content with other entities or people, as long as it follows the legal terms in force
Business rules:	• This rule only applies to teachers linked to an institution and using the platform's institutional plan • Legal responsibility for ownership of the content will be assigned to the institution, as provided for in the terms of use

Source: Author's own work.

All functional requirements categorized as essential priority must be met for the solution to be considered viable. If the solution violates any essential requirement, it will be deemed outside the viability scope for the partner. Violations of other requirements may affect the overall quality of the solution but can be accepted if the global quality remains satisfactory.

3.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements represent attributes related to quality, performance, security, or general constraints within a system (PRESSMAN, 2020). These requirements are crucial for organizations as they address limitations and quality aspects such as execution time, security privileges, and others.

According to Saeedi et al. (2010), quality requirements are key to achieving high performance and ensuring customer satisfaction. In order to preserve the software's usefulness and acceptance by end users, non-functional requirements become essential—always considering the previously identified needs of the target audience.

Each described non-functional requirement includes the following elements: name, description, category, priority, relevant information, and associated business rules.

The adopted categories are:

a) External: Requirements involving the system's interaction with external elements.
b) Product: Requirements related to the internal properties and functionalities of the system itself.
c) Organizational: Requirements related to the organizational environment, the users involved, the context of use, and surrounding operational processes.

Table 7 - Non-Functional Requirements

ID: NFR001	**Requirement Name:** Usability and Intuitive Interface
Description ➞	The platform should have an accessible and easy-to-understand interface, minimizing the learning curve for all user profiles
Category: Organizational	**Priority:** Essential
Information ➞	• Simplified and guided navigation for content creation • Support for screen readers and keyboard navigation • Use of standardized icons and colors • Embedded help texts and interactive tutorials
Business rules:	• The interface should consider profiles with little digital experience
ID: NFR002	**Requirement Name:** Compatibility with Virtual Reality Devices
Description ➞	The application should be compatible with VR and AR devices, especially the VR Meta quest and various cell phones
Category: Product	**Priority:** Essential
Information ➞	• Compatible with headsets such as VR Meta Quest and Google Cardboard and similar • Optimized rendering of light 3D content for cell phones. • 360º experiences available when required
Business rules:	• The experience must be accessible even on low-cost devices.
ID: NFR003	**Requirement Name:** WEB Application
Description ➞	The system will be accessed via a browser on desktop
Category: Organizational	**Priority:** Essential
Information ➞	• Access via modern browsers (Chrome, Firefox, Edge) • Internet connection required to access content
Business rules:	None
ID: NFR004	**Requirement Name:** Database
Description ➞	The system will need a PostgreSQL database to store user information, immersive content, classes and access permissions
Category: External	**Priority:** Essential
Information ➞	• Register: Enter content and user data • Consult / Modify: Updating content and permissions • Consult / Delete: Removing old or incorrect records • Consult / Read: View data by authorized users
Business rules:	None

Source: Author's own work.

All non-functional requirements categorized with essential priority must be fulfilled for the solution to be considered viable. If the solution violates any of these essential requirements, it will be regarded as outside the partner's viability scope. Violations of other non-functional requirements may still lead to a decrease in solution quality but can be accepted if the solution maintains an overall satisfactory level of quality.

4 IMPLEMENTATION OF THE PROPOSED SOLUTION

4.1 USING THE GODOT ENGINE IN VIRTUAL AND AUGMENTED REALITY

4.1.1 TECHNICAL FEASIBILITY OF GODOT FOR VR/AR

The Godot Engine is an open-source game engine that has been consolidating itself as a viable option for academic and experimental projects, despite its support for XR being relatively recent. Compared to established engines such as Unity, Godot presents clear advantages in terms of learning curve and flexibility. The use of GDScript (a language similar to Python) and its node-based architecture make Godot particularly accessible for beginners and students, enabling rapid prototyping without the initial complexity of C# or C++. On the other hand, Unity offers advanced tools and a massive community, which translates into abundant resources and tutorials, albeit with a steeper learning curve. Thus, for the purposes of this Undergraduate Thesis, in which the development time is limited, Godot may prove to be more efficient: an experienced user on the Godot forum reported that Godot "saves a lot of development time and helps maintain motivation" in XR projects.

Among its advantages, Godot is entirely open source (MIT license), allowing deep modifications to the engine and eliminating licensing costs, in addition to being cross-platform (Windows, Linux, Android, etc.). Its modular node-based architecture facilitates the intuitive hierarchical organization of 2D/3D scenes. However, there are limitations to consider: the official XR community in Godot is smaller than that of Unity, and the specific documentation for VR/AR, although growing, is still under development. Recent versions (Godot 4.x) have greatly stabilized support for OpenXR, but some features (particularly ARKit/ARCore) are still not mature within Godot. There are reports that older AR plugins for Godot 3 are outdated and that AR support on mobile (via ARCore/ARKit) is limited, while the current focus of the community is on implementing XR experiences via OpenXR (which includes virtual and augmented reality through passthrough). Conversely, Godot continuously improves its VR-related source code: for instance, version 4.3 redesigned OpenXR frame timing for more precise controller positioning. Overall, Godot offers a lightweight and flexible solution for educational VR/AR prototyping, at the cost of a smaller and still-developing community support.

An important feature of Godot is its ability to integrate 2D content within a 3D environment. By using Viewport or Viewport2D nodes in 3D, it is possible to render 2D scenes (such as graphical user interface panels) on surfaces within the 3D world, facilitating the creation of menus and HUDs in VR. For example, a Viewport2D node can be added inside a MeshInstance and connected to a 2D scene to serve as a texture. In this way, traditional 2D controls (buttons, panels) can be displayed on objects in 3D space and interacted with via the controllers' laser pointer, making the VR UI more intuitive. This technique, addressed in tutorials and in the Godot XR Tools kit, represents a best practice for VR interfaces: it allows prototyping in 2D (testable within the editor) and direct transplant into the VR environment, preserving the scalability and ease of use of Godot's GUI system.

4.1.2 XR PLUGINS IN GODOT: OFFICIAL AND UNOFFICIAL

Godot employs a modular plugin system for XR, encompassing both official solutions and numerous community-developed extensions. The primary standard is OpenXR, supported via an official plugin that abstracts a wide range of devices and runtimes. In addition to this, there are specific plugins: the GodotVR community maintained official drivers for Oculus Rift and OpenVR in its repositories, as well as interfaces for OpenHMD (an open-source project covering various devices). For instance, the driver for Oculus Rift can be downloaded from the Godot Asset Library; however, as reported since 2019, newer headsets such as Quest and Go required adaptations (e.g., GLES2 support) that were still pending. The project has since evolved to gradually replace legacy OpenVR/Oculus support with the standardized use of OpenXR.

Conversely, the community has developed tools to address these gaps. A notable example is the Godot XR Tools, an unofficial plugin maintained by the GodotVR community. This toolkit implements common support scenarios and scripts for XR games (locomotion, interaction, UI, hand tracking, etc.), thereby facilitating development without having to start from scratch. According to the official documentation, XR Tools "implements many of the basic mechanisms found in XR games, from locomotion to interaction with objects and UI," and is compatible with both OpenXR and WebXR. The latest version (XR Tools 4.4.0, February 2025) requires Godot 4.2 or higher, indicating active maintenance and compatibility with the latest engine releases. The kit includes, for example, animated skeletal hands linked to the controllers' trigger/grip inputs, as well as a pointer system for UI interaction. In summary, Godot XR Tools provides a solid and up-to-date foundation for educational VR scenarios, although it is community-maintained and not officially “supported” by the Godot foundation.

Among the official plugins maintained by the Godot team or its sponsors, the Godot Meta Toolkit (released in March 2025) stands out. Maintained by W4 Games and open-sourced under the MIT license, it exposes the Meta Platform SDK (Meta’s network services) and includes tools for exporting to Quest devices. For example, the Meta Toolkit allows automatic configuration of the necessary export options for the Meta Quest’s HorizonOS store. Another official component is the OpenXR Vendors plugin, which is regularly updated (e.g., version 3.1.2 in 2024) to support new features: updates to OpenXR version 1.1.41, enabling hand tracking on Pico devices, and export profiles for Magic Leap 2. These updates attest to strong community support: the ongoing release cycle and integration with new devices (Pico, Magic Leap 2, Quest, etc.) indicate that Godot’s official XR tools are maturing. In parallel, the engine has begun to incorporate support for the XR Editor, enabling users, for instance, to run the editor inside a headset (already functional on Quest 3 and Quest Pro). This demonstrates that the Godot plugin ecosystem for XR has significantly evolved.

Nevertheless, challenges remain. The official documentation for each plugin may be fragmented, and many features require the integration of multiple add-ons (for example, using Godot XR Tools in conjunction with the main OpenXR plugin). Traditional AR support (ARCore/ARKit) remains limited: the consensus within the community is that “AR – Not natively supported, only legacy plugins.” However, Godot already enables AR experiences via passthrough on VR headsets, thanks to the OpenXR standardization of blend modes. In conclusion, when evaluating educational applications, it is advisable to adopt OpenXR as the foundation and supplement it with tools such as XR Tools and Meta Toolkit for specific tasks, bearing in mind that plugins offer ready-made examples (laser menu, hands, etc.) but may require version adjustments and careful reading of release notes.

4.1.3 EXPORT TO VIRTUAL REALITY DEVICES

The process of deploying a Godot VR application to a device such as the Meta Quest essentially follows the same steps as exporting to Android. First, it is necessary to install the Android export template in Godot and configure the Android SDK/NDK (the "Android OpenXR Loader" is recommended for Quest). From that point on, developer mode must be enabled on the headset (via the Oculus app) to authorize the deployment of builds. In the Export menu, a preset for Android is created. Under XR Features, the XR Mode field must be configured as OpenXR, ensuring that the virtual reality runtime is used during export. From this stage onward, the APK can be built normally. Connecting the Quest to the PC via USB and clicking the export button will build and send the application directly to the device (as described in the Godot 3.5 documentation), which also works with Godot 4 by selecting "Export Project" in the editor.

Specifically for Meta Quest 2/3, one must pay attention to the HorizonOS store requirements: modern versions of Godot (4.4+) recognize an export profile for HorizonOS, facilitated by the Godot Meta Toolkit. This plugin adds automatic export options (signatures and settings required by Meta). The Quest 3, being a recent device, is supported in the early releases of Godot 4.4, even allowing the editor itself to run on the headset. In general, builds for Quest are Android ARM64-specific and may require the use of the OpenGLES2 renderer to achieve acceptable performance, given the limitations of mobile hardware. Adjusting the renderer, limiting the framerate, and employing techniques such as foveated rendering are recommended practices to maintain a smooth experience.

For other VR devices, the process is similar: for instance, Pico headsets use OpenXR via Android; the latest version of the OpenXR Vendors plugin includes specific configurations for these devices. AR headsets such as Magic Leap 2 are also supported via OpenXR (with a dedicated export profile). On desktop platforms (HTC Vive, Valve Index, Windows MR), development follows the standard PC workflow (Windows/Linux) with a regular export process; the native OpenXR plugin or wrappers (such as OpenHMD/OpenVR) handle the interface. In all cases, it is necessary only to enable the appropriate Android permissions in the manifest (camera, sensors) and optimize resources: immersive experiences generally require moderate graphics and optimized colliders to maintain 90fps. In summary, exporting for VR in Godot demands attention to XR settings in the project and the specific requirements of each platform, but much of the process can be automated using recent tools (Meta Toolkit for Quest, OpenXR Vendors plugin for others).

4.1.4 THE VR EXPERIENCE IN CHEMISTRY TEACHING

The initial experience developed aims to create an immersive virtual chemistry laboratory suitable for the high school level. The environment simulates a real laboratory: it features a workbench, shelves, a sink, and typical equipment (scales, Bunsen burners, reagent flasks, test tubes, etc.), all modeled in 3D with appropriate textures. Particular attention was given to maintaining realistic scale and plausible lighting in order to maximize the sense of presence. In the virtual laboratory, molecules are represented using a ball-and-stick model for illustration purposes, and chemical compounds are also present in labeled flasks.

The laboratory objects are interactive: each reagent (molecule or solution) is associated with a Godot script that controls its physics and reactions. For instance, when the user places two reagents into the same beaker (using grab-and-release interactions), the simulation can reproduce the corresponding chemical reaction — for example, when mixing potassium permanganate with glycerin, an ignition animation (flame) and heat release occur, reflecting a real exothermic experiment. Particle animations (smoke, bubbles, sparks) and changes in color or brightness provide visual feedback to the user, reinforcing the educational character. Interactions employ Godot's Area and RigidBody nodes: the system detects collisions and interaction signals (picking, proximity) to trigger the reaction process. For example, when a molecule is assembled from smaller parts, the script checks the correct combination and highlights the resulting compound.

To meet high school curriculum requirements, the interface and content are calibrated for the target age group: the selected experiments illustrate basic concepts (simple molecular structures, safe redox reactions, basic stoichiometry). The vocabulary and labels are written in clear Portuguese, and in-game instructions explain the objective of each step (e.g., "Combine these substances and observe the transformation"). Interactive areas feature light gamification elements (such as symbolic scoreboards or achievement markers) to increase engagement without compromising the pedagogical focus. The environment avoids real risks: each experiment that would be hazardous in a physical laboratory (explosions, toxic substances) has been adapted into safe visual versions within the virtual simulation.

The scientific and didactic quality of this experience will be evaluated qualitatively by a professional in the field of chemistry (graduated from USP and active in the industry). Her analysis aims to confirm that the molecular models and selected reactions are chemically accurate and appropriate for the school level. For example, she is tasked with validating that the structure and nomenclature of the molecules correspond to standard tables, and that the programmed reactions demonstrate relevant principles (such as water formation or controlled combustion). From a pedagogical standpoint, she should observe that immersion in VR may enhance intuitive understanding of processes: as other studies suggest, virtual laboratories significantly increase motivation and the retention of chemical concepts. In summary, the first version of the immersive laboratory is intended not only as a technical proof of concept, but also as an educational resource approved by a specialist, demonstrating that Godot can effectively deliver safe and educational chemistry experiences.

4.1.5 FEATURES IMPLEMENTED IN SPRINT 2

In the second development cycle (Sprint 2), interface and interaction functionalities were prioritized. Firstly, an interactive initial menu was implemented: upon launching the VR experience, the user is presented with a floating 2D panel (created using Viewport2D in 3D nodes) containing the options “View Experiences,” “Acquire Experiences,” and “Help.” Selection is performed via the controllers' laser pointer. Utilizing the XR Tools system, the laser pointer was configured to act as a cursor: it emulates a virtual mouse over the 2D viewport interface. Thus, by pressing the trigger button on the controller, the pointer activates the corresponding UI button, initiating the transition to the laboratory environment. This menu takes advantage of the Viewport resource: the 2D interface was designed using standard Godot Control nodes, facilitating layout design, and is displayed on a 3D surface, following best practices for virtual reality interfaces.

Subsequently, the actual virtual laboratory and its interactive elements were created. This involved assembling the 3D laboratory scene (furniture and equipment models) and placing the experimental objects. Each object (molecule, flask, equipment) was developed as a RigidBody3D node with appropriate colliders. For instance, test tubes are RigidBodies that can be grasped by the player; molecules are composed of multiple connected spheres that move with basic physics. To enable manual interaction, the XR Origin design pattern was activated: the “XROrigin” node was defined as the player root, containing the “XRCamera” (first-person view) and two “XRControllers” (virtual hands) as children. This ensures that the user's physical movement in the real space is accurately reflected in the VR environment. The controller hands are equipped with scripts for grasping animation, enabling objects to be picked up and released. The XR Tools kit was employed to streamline this process: for example, each interactive object inherited the XRToolsInteractableBody scripts, allowing automatic response to pointer events or direct grabbing.

Regarding manual interaction animations, visual guides were developed for the user: when holding an object (grip button), the virtual avatar’s arms (represented by virtual hands) display a finger-closing animation (using animation files provided by the kit). This provides immediate and natural feedback. Upon releasing the object, the animation is reversed. These animations were configured in the AnimationPlayer linked to the hand model, triggered by interaction signals from the controller. Different objects have specific colliders to ensure proper grip adherence.

Finally, the XR Origin and interaction pointers were adjusted. The XR Tools system allows switching between direct movement and teleportation; we opted for simple direct movement (repositioning the XR Origin), suitable for small environments. The controllers have a Pointer component installed, which emits a selection ray when the trigger is pressed. This ray interacts both with physical objects (for grabbing) and with invisible colliders that define interactive areas (such as menu buttons or proximity sensors for actions). The layered physical configuration (Physics Layers) ensured that the pointer collides only with the 2D interfaces and intended objects, preventing undesired triggers.

In summary, Sprint 2 resulted in a prototype with only a few functionalities: an active initial VR menu, a 3D laboratory created with modeled and physics-enabled objects, and well-integrated manual interaction mechanisms (grabbing, releasing, selecting). The incorporation of 2D scenarios via Viewport in 3D proved effective for the UI, and the use of XR Origin with a laser pointer enabled natural interactions. Throughout this process, technical challenges such as adjusting fine colliders and synchronizing animations were resolved through constant iterations and the adoption of best practices from the XR Tools kit. The outcome of this sprint provides the foundation to proceed with more experiments and chemical interactions in the next development cycle.

5 PLATFORM AND GENEREAL DESIGN AND DEVELOPMENT

The proposed platform aims to integrate Virtual Reality (VR) and Augmented Reality (AR) resources into high school education, with an initial focus on chemistry, in order to increase engagement and interactivity in the classroom. In the prototype interface developed (in Figma), emphasis was placed on usability: navigation is intuitive and segmented by user profiles (teacher, student, school manager), ensuring that each user accesses the functionalities pertinent to their role. For example, the initial screen for a teacher, upon logging in, presents a dashboard with quick-access panels, including the creation and management of experiences, a content library, and classroom tools. Federated authentication (such as login via Google or Microsoft) is supported to facilitate access, highlighting the concern for user experience from the very first interaction. The visual language of the prototype was designed to be clean and modern, using the name “Immersity” for the platform, and emphasizing on its homepage that “the future of education is immersive,” signaling the pedagogical proposal of transforming traditional lessons into immersive experiences.

From a functional standpoint, the platform offers a comprehensive set of tools for teachers. One of the central components is a drag-and-drop visual editor for VR experiences, which eliminates the need for programming knowledge. In this editor, teachers can select 3D objects from a repository, position them in a virtual scene, and configure interactions (for example, displaying a caption or chemical formula when the student interacts with an object). The prototype illustrates this idea with an editing interface similar to that of a game engine: there is a Scene Hierarchy listing objects (cameras, lights, models, texts, etc.) and an Inspector to adjust properties such as transformations, materials, and behaviors. In a chemistry experience, for instance, the teacher could set up a scene with 3D molecular models and informational panels that appear when a given element of the scene is activated, making the learning of chemical structures much more engaging. Content creation is facilitated by a rich asset library (3D models of molecules, laboratory equipment, explanatory videos, etc.). This library allows access to high-quality educational models and media, ranging from “molecular structures” to visuals of historical phenomena, all at the teacher’s disposal for assembling their virtual lessons.

In addition to creating their own experiences, teachers can share and obtain content through an Experience Store (Marketplace). The “store” allows exploration of VR experiences created by educators from around the world, categorized by subject. This means that a chemistry teacher can find, for example, a ready-made experience on the Periodic Table or Chemical Bonding created by another educator, being able to reuse or adapt it for their class. The Marketplace fosters collaboration and the dissemination of best practices, and includes search and filtering mechanisms (by subject, author, rating) to facilitate the discovery of relevant content. Some experiences may be free while others paid, since the data model provides for a price field in each experience, suggesting the possibility of content commercialization within the platform’s ecosystem. This marketplace-like approach encourages the creation of a community of innovative educators, expanding the reach of high-quality experiences to multiple schools.

To meet the specific needs of each institution, the platform incorporates functionalities aimed at school and class management. There is a “My School” section where managers (or teachers with coordinator roles) can manage the school’s data and its registered users. The prototype, for example, shows information about Colégio Exemplo, including address, contact details, and lists of teachers and students registered on the platform. In this section, it is possible to invite new teachers or students, create new classes, and assign responsible teachers. Each class has a unique code (possibly used for students to enroll in that class) and is associated with a head teacher and a school. The platform allows teachers to add or remove students from their classes and to track which students are enrolled. This management is important to delimit access to experiences: a student only views on their device the experiences assigned to their class by the teacher.

With regard to students, the experience was designed to be as immersive and simplified as possible. Students access experiences preferably through virtual reality headsets, using a dedicated application developed in the Godot engine. The student interface in the VR device shows available experiences (e.g., “Experiences of Chemistry Class 3A”) visually, perhaps as a grid of thumbnails in VR selection mode, or even through a virtual lobby. Once the experience begins, the student is immersed in the 3D environment created by the teacher, being able to interact with objects, observe phenomena on an augmented scale (such as visualizing molecules enlarged and in 3D around them), and carry out guided activities. Student usability focuses on immersion and natural interactivity: for instance, using motion controllers to manipulate virtual chemical objects (such as reagents in a virtual beaker) or freely looking around to inspect a floating molecular model. All of this takes place guided by the design of the experience developed by the teacher, who may include instructions or objectives within the virtual scene itself. Although VR is the primary mode of consumption, the platform also considers AR as an access alternative: for example, students could use a mobile application in AR mode to project 3D models into the real classroom environment (such as viewing a molecule “on top” of the desk via the phone’s camera). This VR/AR versatility broadens accessibility – if the school does not have many VR headsets, tablets or smartphones could be used for AR experiences, albeit with reduced immersion.

In terms of user experience (UX), the platform seeks to integrate the traditional flow of the lesson with new technologies in a seamless manner. The teacher remains at the center of lesson planning, using the platform as a didactic tool. The learning curve of the interface was intentionally kept low: familiar elements from other tools (clear menus, organized panels, drag-and-drop) ensure that teachers can start producing content quickly, even without prior experience in 3D development. Likewise, for students, the proposal is that accessing an experience should be as simple as opening a textbook—with the difference that the “book” now completely surrounds them. The high interactivity seeks alignment with constructivist learning theories, in which students learn better through active experimentation with the content. Viewing a simulation of a chemical reaction in VR, for example, may spark questions and curiosity much more effectively than merely observing an image on the board. Thus, the platform intends to complement traditional teaching, not replace it: the teacher can introduce a concept in a lecture and subsequently lead students into an immersive activity that reinforces that concept, increasing engagement and knowledge retention.

5.1 SOFTWARE ARCHITECTURE

The software architecture of the platform was designed to be scalable, secure, and modular, leveraging Amazon Web Services (AWS) cloud services to support the distribution of VR/AR content to multiple users. According to the attached architecture diagram, the solution adopts a clear division into layers: frontend (web client and VR applications), backend (API and server-side business logic), and storage (database and media file repository).

Cloud Infrastructure (AWS): The entire infrastructure resides on the AWS cloud, taking advantage of managed services to reduce the complexity of server management. The web interface (frontend) is an application built in Next.js (React), deployed statically and distributed through Amazon CloudFront, a content delivery network (CDN). CloudFront serves the frontend files (HTML, CSS, JavaScript, static assets) globally, ensuring low latency for users in different locations, and can also route API traffic to the appropriate servers. This means that browser requests to /api/... can be forwarded by CloudFront directly to the backend, while requests for static resources are immediately served from the CDN cache. Behind CloudFront, the components reside inside a Virtual Private Cloud (VPC), isolating them in private subnets for greater security. The application backend is implemented in ASP.NET Core (indicating a RESTful .NET API) and runs in Docker containers orchestrated by AWS Elastic Container Service (ECS), using the Fargate modality (which eliminates the need to directly manage instances). This choice allows business logic to be easily scalable: new container instances can be launched as user load increases, maintaining performance during peak hours. The backend communicates with a managed Amazon RDS (PostgreSQL) database to read/write persistent data, such as information on users, experiences, classes, etc. RDS ensures reliability (with automated backups, optional multi-AZ replication) and consistency of academic and content-related data.

Asset Storage and Distribution: Large files such as 3D models, textures, videos, or audio used in VR/AR experiences are not served directly by the backend, but instead stored in Amazon S3 (Simple Storage Service). S3 is ideal for storing binary objects in a scalable and highly durable manner. Each asset receives a unique identifier and is stored in a specific bucket; references to these files are maintained in the database only as paths or keys (for example, an s3_key for the bucket location). To distribute this content efficiently and securely, the architecture makes intensive use of pre-signed S3 URLs. The backend generates these temporary URLs whenever an authorized user needs to upload or download an asset. For example, when a teacher uploads a new 3D model through the web interface (adding an asset to the library or scene), the client first requests a pre-signed upload URL from the backend; the file is then sent directly from the browser to S3 using this URL. This process offloads heavy file transfers from the server, leveraging S3's optimized infrastructure for data reception, while maintaining access control (the URL expires after a short period and only allows the specified operation). Similarly, when a student using a VR headset needs to download assets for an experience, the VR application requests pre-signed download URLs for each required file and then retrieves them directly from S3. Since CloudFront can be configured as an edge cache for the S3 bucket, downloads of popular assets (e.g., a widely used model in several experiences) can be delivered even faster via a nearby CDN node. This S3 + CloudFront combination ensures fast delivery of static content, essential in VR where model and texture sizes can be large.

Authentication and Authorization: Access security is handled by Amazon Cognito, which provides user management and authentication. Each user (teacher, student, etc.) upon registering in the platform has their credentials validated and stored by Cognito, which manages details such as secure password storage and integration with social logins. Cognito simplifies the implementation of features like login/signup, email verification, and password resets, while offering compliance with security standards without requiring developers to implement them from scratch. When a user logs in (via the web application or VR app), Cognito issues JWT authentication tokens, which are sent in each request to the backend. The .NET API validates these tokens—ensuring that only authenticated users can access sensitive endpoints—and can extract information from them such as user ID and permissions (role). The integration between Cognito and other AWS services also facilitates, for example, the generation of pre-signed S3 URLs with the correct temporary credentials, or differentiated access to resources according to the user profile (a student can only obtain URLs for assets of experiences assigned to their class, for instance). Furthermore, Cognito can be used to distinguish access levels: for example, a platform administrator (user_role = admin) could have a token with a scope allowing access to administration endpoints that a regular teacher cannot access. The layered design, combined with JWT tokens, also enables the possibility of integrating other clients in the future (e.g., a standalone AR mobile application) while securely reusing the same backend.

Communication between Web Clients, VR, and Server: Interaction flows vary according to the client type, but both converge on the backend. The web client (teacher's browser) communicates with the backend via HTTPS REST calls (or GraphQL, if implemented), sending authentication tokens in headers. Operations such as creating experiences, editing, assigning to classes, library queries, etc., are API requests, which in turn execute business logic (e.g., saving data in PostgreSQL, calling email services, etc.). The VR client (Godot application running on the student's headset) typically does not have a rich web interface but still needs to interact with the server: to authenticate the student (likely via Cognito, entering username/password or scanning a login QR code) and to obtain the list of experiences assigned to their class and their details. The VR app then uses the same APIs (or a subset of them) to retrieve experience data and assets. Since it runs on a possibly standalone VR device (such as an Oculus Quest), the app needs to optimize network usage—for example, downloading only what is necessary and possibly pre-downloading assets before class begins (local caching) to avoid latency during the experience. The cloud architecture supports such scenarios and could even use notifications (via MQTT or WebSocket) if real-time updates were required, although the provided diagram primarily indicates pull communication (the device queries when ready).

Layer Separation and Technical Justifications: The choice of a distributed architecture with distinct layers brings several advantages. The Next.js frontend layer allows the creation of an interactive and dynamic UI, running mostly on the client, which is ideal for editing tools (drag-and-drop, 3D preview in the browser) without overloading the server. At the same time, Next.js enables efficient rendering and could even generate static pages for public content (such as the landing page "The future of education is immersive…"), improving SEO and initial performance. The .NET Core backend layer was likely chosen for its robustness, high-performance support, and alignment with engineering practices (including easy integration with PostgreSQL and AWS SDK). The use of containers (ECS/Fargate) provides automatic scalability: in a scenario of widespread adoption, where hundreds of students may access experiences simultaneously, additional backend instances can be launched on demand, ensuring fast API responses. The PostgreSQL data layer was justified by the need for relational consistency—multiple interconnected entities, complex transactions (e.g., enrolling a student requires checking school, class, etc.)—and by the team's familiarity with SQL, in addition to advanced features such as JSONB (used to store semi-structured scene data). As previously mentioned, S3 and CloudFront were natural choices for delivering heavy content globally, both being highly optimized AWS services that guarantee reliability and speed in file delivery, which is crucial for preventing delays in VR demonstrations. Finally, the adoption of AWS Cognito and other managed services reflects a decision to reduce the burden of implementing security and authentication infrastructure, allowing the team to focus on developing educational functionalities. In short, the software architecture was designed to support a production-scale product, with the potential to serve multiple schools and thousands of users, maintaining appropriate isolation (each school only sees its own data) and flexibility to evolve (for example, adding new media types or easily migrating components due to decoupling).

5.2 DATA MODEL

The system's data model was conceived in a relational manner, using PostgreSQL tables to represent the platform's core entities and their relationships. According to the provided database diagram, the following main tables stand out: Users, Schools, Classes, Experiences, Assets, as well as relationship tables such as ClassEnrollments, ClassAssignments, ExperienceAssets, and CustomContentRequests. Together, these support the multiplicity of user profiles and their interactions, ensuring referential integrity and allowing efficient queries for the application's use cases.

The Users table stores the data of all platform users, whether teachers, students, administrators, etc. Each user has a unique identifier (UUID) and basic information such as name, email, and password (stored as a hash). An important field in Users is role, which indicates the user's profile, restricting their permissions within the system. This field is defined from an enumeration user_role, whose possible values include admin, school_manager, teacher, student, and creator. Thus, the data model supports multiple user profiles:

a) Admin: global platform administrator, with broad permissions (e.g., manage schools and content at a systemic level).
b) School_manager: manager of a specific school, responsible for administering users (teachers/students) of that institution.
c) Teacher: teacher, with access to create experiences, manage their classes, and assign activities.
d) Student: student, with access only to consume the experiences assigned to them and eventually view their progress.

e) Creator: a specialized content creator (possibly a member of the platform team or partner) who can fulfill custom content requests made by teachers.

Each user is associated (school_id) with a school in the Schools table, except possibly global administrators or creators who may not belong to any specific school (in which case school_id could be null or generic). The Schools table, in turn, stores institutional information (name, tax ID, address, etc.) and serves to delimit a collaborative space: teachers and students linked to the same school can interact within the same classes and share internal experiences, while users from different schools remain isolated (except via the public marketplace). This separation is crucial for organizational scalability—a single system instance can host multiple schools without mixing their data, enabling a multi-tenant model.

The Classes table represents the classes within a school, each identified by a UUID and containing attributes such as class name (e.g., "Chemistry 3A"), reference to the school it belongs to, and the teacher responsible for teaching it. There is also a class_code field, a short code (8 characters) likely used for students to enroll in the class or for inviting them in a more user-friendly way than using the UUID. The relationship between Classes and Users (teachers) is many-to-one: one class has exactly one responsible teacher (teacher_id), but a teacher can teach several classes. Students participating in a class are not listed directly in Classes; instead, the many-to-many relationship between students and classes is captured by the ClassEnrollments table. Each record in ClassEnrollments consists of class_id and student_id, indicating that a given student is enrolled in a specific class, and also includes the enrollment date. This design allows a student to belong to multiple classes (for example, if the platform is used in several subjects, the same student may have ClassEnrollments with the Chemistry class, another in Mathematics, etc.), and conversely, each class may have many students. The separation into ClassEnrollments facilitates operations such as listing all students in a class or all classes of a student, in addition to maintaining history (it could be extended with enrollment status, grades, etc., if necessary).

At the heart of the system is the Experiences entity, which models an educational VR/AR experience created on the platform. Each experience has fields such as title (title), description (textual description), creator_id referencing the user (typically a teacher or content creator) who created it, along with important metadata for its lifecycle and access. Among these metadata, the following stand out:

a) status: indicates the state of the experience (e.g., draft, published, or archived, as defined in the experience_status enumeration). This allows the teacher to develop an experience in stages, publishing it only when ready for students, and eventually archiving it when no longer relevant.
b) visibility: defines the visibility/sharing of the experience, according to the experience_visibility enumeration. Values may be public (available to any user on the platform—appears in the public Marketplace), private (visible only to the creator teacher and those they explicitly share with), or school_shared (shared within the

creator's school, meaning other teachers in the same school can use it). This distinction enables fine control over who can discover and use the experience.

c) price: a monetary value indicating whether the experience is commercialized and at what price (in a specific currency, e.g., Brazilian reais). If set (> 0), the experience is presumed to appear in the Marketplace requiring purchase; if zero, it would be free.

d) subject: the subject or thematic area of the experience, defined by the subject_type enumeration. Values cover a range of curriculum subjects (History, Geography, Chemistry, etc., including Chemistry, Physics, Biology, and others).

e) scene_data: this field, stored in PostgreSQL's JSONB (binary JSON) format, contains the structural description of the scene and its objects. Instead of normalizing all scene elements into multiple tables (which would be highly complex and rigid given the diversity of possible objects), a flexible JSON field was chosen, where properties such as object lists, positions/rotations, and specific parameters are recorded (e.g., "object X is a given 3D model, with interaction type display text Y when clicked"). This approach facilitates the evolution of the scene editor—new object types or properties can be added without altering the database schema, and JSONB even allows specific queries within the field if needed, as well as efficient indexing by PostgreSQL for certain attributes.

The Assets table complements the experience model by storing the media assets used in these experiences. Each asset also has its UUID, original filename, S3 key (s3_key), and a type (file_type) defined by the asset_type enumeration. The types cover common 3D and multimedia files: for example, model_gltf (3D model in glTF format, widely supported by engines such as Godot), image_jpeg / image_png (static images), video_mp4 (video), and audio_mp3 (audio). This indicates that the system supports a variety of media to enrich experiences—from interactive 3D models to explanatory videos or audio narrations within the virtual environment. The presence of uploader_id in Assets makes it possible to track who uploaded a given resource (e.g., a teacher uploaded a specific model) for ownership and possibly management purposes (an unused asset could be removed by its owner or by admins). Importantly, the files themselves reside in S3; this table serves to index them and logically associate them with experiences.

To relate Experiences and Assets, the model includes the ExperienceAssets table, which is a many-to-many join table between experiences and assets. An experience can use several assets (a chemistry scene could have models of different molecules and a demonstration video), and conversely, the same asset (such as a water molecule model) can be reused in multiple experiences. Each entry in ExperienceAssets consists of experience_id and asset_id, indicating that a given asset is part of the experience's scene. This normalization avoids file duplication in cases of reuse and facilitates updates—if a model is improved, all experiences that reference it automatically gain access to the new version, without the need to modify each experience individually (as long as the s3_key is properly updated, maintaining the reference).

The platform provides for the distribution of experiences to classes through the ClassAssignments table. This table indicates that a certain experience was assigned (assigned_at) to a given class (class_id), usually by a teacher (assigned_by_id). In practice, this implements the functionality of a teacher selecting an experience (created by them or obtained from the marketplace) and making it available to one of their classes. Each entry in

ClassAssignments means “Class X has access to Experience Y as of a certain date, assigned by Teacher Z.” With this structure, queries such as “which experiences does Class 3A currently have available?” or “which classes has this teacher already assigned the Acid-Base experience to?” become straightforward. When a student logs in, the system can retrieve all ClassAssignments where class_id corresponds to a class in which that student is enrolled (via ClassEnrollments), thus obtaining the list of experiences they can access. This approach also makes simple tracking possible—although there is no field for completion status per student here, the model could be extended for that purpose (for example, with a StudentExperienceProgress table), but that goes beyond the initial scope.

Finally, an interesting differential of the platform is the inclusion of personalized content requests through the CustomContentRequests table. This table stores requests for new content made by teachers (requester_id and description fields), eventually directed to the platform’s creator team. For example, if a teacher wants a VR simulation of a specific chemical experiment that they cannot find in the library, they can formalize a request describing their need. The record includes school_id (perhaps for context or approval by the school manager) and a status indicating the progress of the request, according to the request_status enumeration: pending (under review), in_progress (under development), completed (fulfilled, with the content already produced), or rejected (declined/not feasible). Although the database diagram does not specify who fulfills these requests, it can be inferred that creator or administrator users would have access to manage them, possibly linking a new Experience to the request once completed. This functionality demonstrates the platform’s commitment to evolving according to faculty demands, enabling continuous validation of tools and expansion of resources based on direct user feedback. In summary, the data model was carefully designed to support the various flows (creation, sharing, assignment, request) while maintaining consistency—for example, ensuring that only teachers can create experiences (controlled by the application through the role field, not referential integrity), or that a student cannot access experiences from another school (since the relationship between Classes, Schools, and Assignments enforces that scope).

5.3 CLASS DIAGRAM

The class diagram of the application, developed in UML, translates the data structures and business rules into an object-oriented model used in the implementation (especially in the backend layer and possibly partially in the frontend). It includes the main classes corresponding to the educational domain entities and their relationships, as well as enumerations that define important constants. The following describes the object-oriented structure, focusing on the main classes (User, School, Class, Experience, Asset, CustomContentRequest) and how their methods and relationships were designed to promote encapsulation and reusability.

The central class is User, representing all types of system users (teachers, students, etc.). Its attributes include id, full name, email, password (hash), and role (of type enum UserRole). The diagram highlights that User is associated with student enrollments – that is, there is a relationship between User (when in the student role) and ClassEnrollment, indicating the classes in which that student is enrolled. There is also a relationship between User (when teacher) and Class, denoted by “is taught by (teacher)”: a class is taught by a teacher (User). This reflects multiplicity: a teacher may teach several classes, but each class has only one responsible teacher. In code, the Class class would likely have a property such as teacher: User. The User class also aggregates a collection of created experiences (when the user is a teacher or creator), although in the diagram this link appears via methods. The illustrated methods of the User class include: register(), login(), updateProfile(), createExperience(), uploadAsset() and enrollInClass(classCode). These methods encapsulate important user-related functionalities of the system. For example:

a) register() and login(): most likely interact with the authentication service (Cognito) to create a new account or validate credentials, and also create corresponding entries in the Users table of the database, possibly returning an access token. Having them as methods suggests that there is domain logic (such as validating email format, encrypting the password, etc.) concentrated there.
b) updateProfile(): allows the user to change their data (name, password, etc.) with the appropriate business validations (for example, preventing duplicate emails).
c) createExperience(): would be used by a teacher or creator to instantiate a new experience (Experience object), filling in basic fields and perhaps establishing the authorship relationship (setting creator_id equal to the current User id). This method could encapsulate logic such as “a student cannot create experiences” (checking the user’s role), ensuring that only permitted roles execute the creation.
d) uploadAsset(): similar to the previous one, would involve requesting a pre-signed URL and registering a new Asset object associated with the user. Having it as a User method emphasizes that every upload is linked to a user (uploader_id) and possibly that different roles may have restrictions (e.g., perhaps students cannot upload, or only certain user types can).
e) enrollInClass(classCode): represents the action of a student joining a class. Internally, it would search for a Class instance whose code matches the provided classCode and create a new ClassEnrollment linking this User (as a student) to that class. By placing this operation in User, it ensures that enrollment logic (such as checking if the code is valid, or if the student is not already enrolled) remains cohesive. Alternatively, it could be in Class (addStudent(user)), but having it in User suggests the student’s perspective of joining the class.

The School class reflects the school entity and encapsulates its relationship with users and classes. A School aggregates many users – the diagram notes “a School has many Users (managers, teachers, students)” – and also contains many classes. The School class has attributes such as id, name, CNPJ (corporate registration number), address, etc., corresponding to the Schools table. Its methods include: addUser(user), createClass(teacher, name), getTeachers() and getStudents(). These methods provide an interface for managing elements within the school context:

a) addUser(user): could perform procedures to link an existing user to that school, or create a new user already associated with it. It may be used by an admin or school_manager when registering new teachers and students, ensuring that the school_id field of User is filled correctly and that business rules (for example, a school can only have one user with a given email) are respected.
b) createClass(teacher, name): would instantiate a new Class under that school, defining the responsible teacher and other basic attributes. Here, encapsulation ensures that every new class is already associated with a school and has a valid teacher. It could also automatically generate a unique class_code.
c) getTeachers()/getStudents(): would return lists of users from that school filtered by their role (role = teacher or role = student). Thus, the School class concentrates the logic to obtain these sets, instead of scattering queries across the system, which is useful to, for example, display the "My School" page with all members.

The Class class represents school classes and, consistently with the Classes table, has attributes id, name, code, creation/update dates and references to the teacher and school (in the diagram, the relationship with School is indicated textually/implicitly). In terms of relationships, a Class contains collections of ClassEnrollment (student enrollments) and ClassAssignment (assigned experiences) – in the diagram, Class contains "enrollments (student)" and "assignments." This suggests that internally the Class class may have properties such as students: List<User> (or enrollments: List<ClassEnrollment> for details) and assignments: List<ClassAssignment>. However, depending on the modeling choice, it may not expose user lists directly, but rather methods for manipulation. Indeed, the Class methods listed include: addStudent(student), removeStudent(student), assignExperience(experience) and getAssignments(). These methods implement class management logic:

a) addStudent(student): would create a new enrollment (ClassEnrollment) linking the student to the class, probably checking whether the student is not already enrolled and belongs to the same school before confirming.
b) removeStudent(student): would similarly terminate an enrollment – possibly marking it as removed or deleting the ClassEnrollment record – thus removing the student from the class.
c) assignExperience(experience): would create a ClassAssignment, i.e., record that the class now has that experience assigned. Internally, it would ensure there is no duplication (not assigning the same experience twice) and fill fields such as assigned_by (probably with the class teacher, assuming only they or a manager can assign).
d) getAssignments(): would return the experiences currently assigned to the class, facilitating queries for teachers or for displaying to students.

The Experience class represents immersive experiences themselves. Its attributes correspond to those already described in the Experiences table: id, title, description, status (enum ExperienceStatus), visibility (enum ExperienceVisibility), price, subject (enum SubjectType), scene data (possibly stored as an object or analogous to JSONB), and timestamps. The methods include: saveDraft(sceneData), publish(), archive(), addAsset(asset) and possibly assignToClass(class) (the textual diagram was truncated, but this is inferred). These methods promote encapsulation of business rules related to the lifecycle of experiences:

a) saveDraft(sceneData): saves/updates the experience content (object positions, configurations, etc.) in draft state, without publishing it.
b) publish(): changes the status to published, making it available according to its visibility, possibly checking prerequisites (such as filled fields, valid scene, etc.).
c) archive(): changes the status to archived, signaling that the experience should no longer be used in new classes or listings.
d) addAsset(asset): relates an asset to the experience, likely creating an ExperienceAsset association.
e) (possible) assignToClass(class): although assignment exists in Class, a dual method may exist here for convenience, initiating the assignment from the experience's perspective.

The Asset class represents media assets and resources. Its attributes include id, filename, S3 key, type (enum AssetType), and upload date. A notable method is getPresignedUrl(), suggesting that the class itself may provide the temporary download (or upload) link for that resource, encapsulating the call to the storage service.

The CustomContentRequest class models customized content requests made by teachers. Its attributes include id, request description, status (enum RequestStatus), and dates. It also references the requester (User) and possibly the School. Its associated methods are: approve(), reject(), complete(). These encapsulate the request approval workflow.

The diagram also includes enumerations (UserRole, ExperienceStatus, ExperienceVisibility, SubjectType, AssetType, RequestStatus) that are used in attributes, ensuring clarity and type safety.

From an OO design perspective, aggregation and composition are employed to model containment relationships. For example, a Class composes its ClassEnrollments and ClassAssignments, meaning they conceptually do not exist without the class. Conversely, relationships such as User–School or User–Experience (creator) are weaker (aggregation). Encapsulating operations within class methods centralizes business rules and ensures maintainability.

In summary, the class diagram reflects a software design organized around educational domain entities, using object-oriented concepts to align with the relational data model and the platform's required operations. This model facilitates translating high-level needs (such as "a student enrolls in a class and accesses experiences") into structured code, making implementation and future maintenance more straightforward.

5.4 INTEGRATION AMONG COMPONENTS

With the platform's components defined—interface/prototype, cloud architecture, relational data model, and domain classes—it is essential to understand how they connect throughout the platform's lifecycle. The integration encompasses the entire process, from the creation of a VR experience by the teacher to its consumption by the student on a VR device, including data flows between frontend, backend, database, AWS services, and the Godot engine responsible for running the experiences. The following sections describe this complete use case step by step, illustrating the collaboration among components at each stage.

Creation of an Experience by the Teacher: The process begins in the web interface (frontend) when a teacher decides to create a new immersive experience for their class. Once authenticated, they access the "My Experiences" section and select the option to create a new experience. The frontend loads the 3D editor (possibly implemented with support from the Godot engine or WebGL libraries), which presents the teacher with an empty environment and construction tools. As the teacher adds elements—for example, inserting a 3D model of a molecule, positioning explanatory texts, configuring lights, and defining interactions—the web application updates an internal representation of the scene (in JSON format).

These resources may be retrieved from the assets library: the teacher searches for the desired model (e.g., a glucose molecule) and inserts it. If the asset already exists in the system (for instance, a publicly available model), the frontend references it directly; if it is a new asset (the teacher imports a file from their computer), the frontend performs the upload procedure via an S3 pre-signed URL, as described in the architecture. At this point, AWS S3 integration is triggered: the frontend requests an upload URL from the backend (API), the backend uses the AWS SDK to obtain the URL from S3 and returns it, after which the file is sent. The backend then creates a record in the Assets table with the file's metadata and associates it with the uploader (the teacher). Once all desired objects have been added to the scene, the teacher clicks "Save." The frontend then sends an HTTP request (POST/PUT) to the backend containing the metadata of the experience: title, description, subject area, and the JSON scene structure (all object and asset configurations). On the backend, the corresponding controller invokes domain methods (e.g., User.createExperience() or a service function that instantiates an Experience object and persists it). This operation records a new entry in the Experiences table (with the default status "draft") in PostgreSQL, including the JSON scene and creator reference, as well as creating entries in ExperienceAssets linking the assets used. The result (success, ID of the created experience) is returned to the frontend, which then notifies the teacher that the experience has been saved as a draft.

During editing, the teacher may preview the experience to see how it would appear in VR. This preview could take place either directly in the browser (using a controllable 3D camera) or on a VR device, if available. In the case of a browser preview, the frontend would need to load the models and render them in 3D—possibly using Three.js or a Godot WebAssembly component. If the teacher uses a VR headset connected to the computer (e.g., via WebXR or a native Godot application), they may test the experience firsthand prior to publication.

Publication and Distribution of the Experience: Once satisfied with the draft, the teacher proceeds to publish the experience and distribute it to their class. In the interface, they change the status to “Publish,” triggering in the backend (via API) the method Experience.publish() or its equivalent, which updates the status to “published” and possibly sets the desired visibility (e.g., private or school-shared, if it is curriculum-specific content). To ensure student access, the teacher navigates to the class management section (“Classes” or “My Classes”) and assigns the experience to the intended class. This likely involves selecting the newly created experience, choosing the class (e.g., Chemistry 3A), and confirming the assignment. The frontend then performs a POST request to the ClassAssignments API, resulting in the creation of a record in ClassAssignments linking the class and the experience. The assigned_by field is filled with the teacher’s ID and assigned_at with the current timestamp. From this moment onward, any student enrolled in class 3A (either already enrolled or enrolling later) is granted access to the experience.

Here, integration between models and business logic becomes evident: the Class object (representing class 3A) internally executes something equivalent to assignExperience(experience), and the domain Experience object records that it was assigned. The system might also notify the students of the new activity (via email or push notifications—possibly through Amazon SNS or WebSocket—although not explicitly described). In simpler scenarios, the notification occurs when the student opens the app.

Access by the Student in VR: At the scheduled time, students put on their VR headsets and open the platform’s application, developed with the Godot engine. On the initial screen (in VR), the student is prompted to log in, which may be implemented via a simple 2D interface or an auxiliary device (some VR apps allow login via an external browser using a code). Authentication is performed with credentials stored in Cognito. The Godot application communicates with the backend (via HTTPS requests, possibly using Godot’s HTTPRequest class) to validate login. Once authenticated, a JWT token is issued and stored temporarily in the app. From this point on, the app can access secure APIs.

The student’s first step is typically retrieving the list of available experiences. With the token, the app makes a GET request to an endpoint such as /me/assignments. The backend decodes the JWT to identify the user and their role (student), queries the ClassEnrollments table to find the student’s classes, and cross-references with ClassAssignments to retrieve assigned experiences. If the student is enrolled in Chemistry 3A, the backend locates the record linking that class with the published experience. It returns a list of experiences (possibly one or more), including metadata such as title, description, cover image, and an identifier. The app displays this list in the VR lobby, allowing the student to select the desired experience (e.g., “Visualization of Chemical Bonds,” as named by the teacher).

At this stage, the VR app must load the experience. The process is as follows: the app issues a GET request /experiences/{id} for the detailed content. The backend verifies permissions, retrieves the Experiences record (including the JSON scene_data and asset references via ExperienceAssets), and prepares a response containing the scene payload. This may include the JSON description directly or an optimized version, along with either pre-signed download URLs for the necessary assets or the asset keys requiring further requests. The more efficient solution involves the backend generating pre-signed URLs for all required assets and embedding them in the response. The Godot app receives this payload, downloads assets directly from S3/CloudFront using the pre-signed URLs, and dynamically loads them into memory.

The app then reconstructs the scene in VR using Godot: objects are instantiated, positioned according to the JSON, and configured with properties (e.g., an interaction script that displays text when clicked). This requires tight integration between the web editor and the Godot runtime, both relying on a shared scene description format (scene_data). Once the scene is built, the student is fully immersed in the VR experience, interacting as designed by the teacher (e.g., exploring molecules, manipulating virtual reagents).

Feedback and Post-Experience Interaction: After the student completes the activity, feedback may be recorded. For example, the VR app may send a POST request /experiences/{id}/complete, logging the completion. Although the current data model lacks explicit tracking of student progress, such functionality could be easily extended with a StudentExperienceStatus table. Alternatively, qualitative feedback could be collected through a survey or evaluation form, either integrated into the platform or via classroom discussion. This loop allows teachers to improve their experiences iteratively.

Custom Content Requests: Another integration scenario involves teachers requesting specific content. Suppose a teacher requires a 3D model of a specific chemical compound not available in the library. They submit a request via the frontend, which the backend records in CustomContentRequests (status: “pending”). An administrator or content creator reviews and, if viable, approves it (approve() method). The creator then develops the requested content (using tools such as Godot or Blender), creates the corresponding Experience or Asset, and marks the request as “completed.” The teacher is notified and gains access to the new content, thereby enriching the platform’s library.

Role of the Godot Engine: Throughout this process, the Godot engine plays a critical role as the runtime environment for VR/AR experiences. While most of the platform operates on the web (editing, management, and distribution), the immersive content is delivered via Godot-based applications on final devices (VR headsets, and possibly AR-enabled mobile devices). The choice of Godot stems from its open-source nature, lightweight performance, and VR support through OpenXR APIs, enabling broad compatibility across devices without licensing costs. Integration is facilitated by standardized formats (e.g., glTF for 3D models, PNG/JPEG for textures, MP4 for videos, MP3 for audio), ensuring interoperability across the web editor, backend, and Godot runtime.

Summary of Technical Flows: Integration can be summarized in four primary workflows:

a) Editing Flow (Teacher-Web): Browser ↔ Frontend (Next.js) ↔ Backend API (ECS/.NET) ↔ Database (RDS) & Storage (S3).
b) Publishing/Distribution Flow: Teacher-Web ↔ Backend API ↔ Database.
c) Consumption Flow (Student-VR): Godot VR App ↔ Backend API ↔ Database & S3.
d) Feedback/Request Flow: Teacher-Web ↔ Backend API ↔ Database ↔ Creator (via notifications).

Each of these workflows was designed to minimize friction: CDN and pre-signed URLs for efficient asset delivery, structured JSON for scene descriptions, JWT tokens for unified authentication across clients, and domain-level encapsulation to enforce business rules.

The platform thus emerges as a cohesive ecosystem in which teachers can easily transform pedagogical ideas into shareable virtual experiences, supported by a robust infrastructure for distribution and collaboration. This design enriches the teaching process, particularly in chemistry education, where making the invisible (atoms, molecules, reactions) visible through VR can substantially enhance students' understanding of abstract concepts.

6 CONCLUSIONS

To-do

BIBLIOGRAPHICAL REFERENCES

To-do

APPENDIX A

To-do