MEJORA DEL APRENDIZAJE BASADO EN LA LÓGICA: APLICACIÓN DE GUESS-18 PARA LA EVALUACIÓN DE LA EXPERIENCIA DEL USUARIO

Enhancing Logic-Driven Learning: Applying GUESS-18 for User Experience Evaluation

David Bonilla Carranza [1]	jose.bcarranza@academicos.udg.mx
Adriana Peña Pérez Negrón [2]	adriana.ppnegron@academicos.udg.mx
Mirna Muñoz Mata [2]	mirna.munoz@cimat.mx

[1] Universidad de Guadalajara (Centro Universitario de Ciencias Exactas e Ingenierías)
[2] Centro de Investigación en Matemáticas (Unidad Zacatecas)

RESUMEN

Este artículo detalla un estudio que examina la experiencia del usuario en un juego educativo de ingeniería de software a través de la Escala de satisfacción de la experiencia del usuario del juego (GUESS-18). En él participaron 111 estudiantes de ingeniería informática que jugaron un juego diseñado para mejorar sus habilidades de software a través de tareas interactivas. El GUESS-18 ayudó a medir el impacto del juego en la satisfacción del usuario, mostrando mejoras notables en la capacidad de resolución de problemas y el compromiso de los estudiantes. El estudio también destaca el potencial de GUESS-18 como una métrica valiosa para contextos educativos, lo que indica su utilidad más amplia más allá del simple entretenimiento. Los hallazgos positivos sugieren nuevas direcciones para incorporar herramientas de experiencia de usuario en software educativo, con el objetivo de mejorar la calidad de la tecnología educativa en el campo de la ingeniería informática.

Palabras Clave: GUESS-18, experiencia de usuario, juegos educativos, educación en ingeniería de software, aprendizaje interactivo, diseño de juegos.

ABSTRACT

This article details a study that examines the user experience in a software engineering educational game through the Game User Experience Satisfaction Scale (GUESS-18). It involved 111 computer engineering students who played a game designed to improve their software skills via interactive tasks. The GUESS-18 helped measure the game's impact on user satisfaction, showing notable gains in students' problem-solving abilities and engagement. The study also highlights the potential of GUESS-18 as a valuable metric for educational contexts, indicating its broader utility beyond just entertainment. The positive findings suggest new directions for incorporating user experience tools into educational software, with the goal of enhancing the quality of educational technology in the field of computer engineering.

Keywords: GUESS-18, User Experience, Educational Gaming, Software Engineering Education, Interactive Learning, Game Design.

Fecha de Recepción: 23/nov/2023. **Fecha de Aceptación:** 01/mar/2024 **DOI:** https://doi.org/10.47187/perspectivas.6.2.225

I. Introducción

Technological advances led to changes in educational and assessment processes, especially in the field of programming and Computer Science. Traditional assessments based on sequential questions may not accurately reflect students' abilities, because external factors might influence their answers. With the aim to find a didactic and playful way to assess programming students' skills, use video games has been proposed as an assessment approach [1].

Software engineering is evolving rapidly, with a growing emphasis on interactive and hands-on learning approaches. Video games may be the answer to better assessment methods. "Logic-Driven Learning,"[2] an interactive video game designed for this purpose, embodies this shift by offering a platform where users can engage with software engineering concepts through gameplay. This game aims to enhance students' programming logic and understanding of software engineering principles in an engaging and dynamic environment. The importance of such tools in education is increasingly recognized, particularly in addressing the high attrition rates in software engineering courses, which often stem from a lack of practical, real-world application of theoretical knowledge.

To effectively measure and enhance the educational impact of "Logic-Driven Learning," it is crucial to assess not only the learning outcomes but also the user experience and satisfaction. This is where the Game User Experience Satisfaction Scale (GUESS),[3] a comprehensive tool for evaluating user satisfaction in gaming environments, becomes integral. The original GUESS, with its 55 items, provided an extensive framework for such assessments. However, its length posed practical challenges in situations requiring rapid and repeated evaluations, such as in iterative game design. Consequently, the development of GUESS-18, an abbreviated version of the original scale, offers a more practical solution without compromising on comprehensiveness or reliability.

In this study, we present how "logic-based learning" is developed and we have used GUESS-18 tests to evaluate how game design and content influence user experience and learning outcomes. This integration aims to provide a dual perspective: firstly, evaluating the game's effectiveness in imparting software engineering skills and, secondly, understanding the elements of user experience that contribute to or detract from the educational value of the game. By applying GUESS-18, we aim to validate this tool in a new context – educational gaming for software engineering – and to gather insights that could guide future enhancements to the game's design. This approach aligns with the emerging trend in educational technology that emphasizes the importance of user experience in learning outcomes, aiming to create more effective, engaging, and user-friendly educational tools.

II. Related work

In this section are analyzed related works from three perspectives included in this research: (A) programming centered systems, (B) programming focused video games, and (C) video games focused on teaching software engineering good practices.

A. Programming-centered systems

Currently there are several web pages that consist of solving logical algorithms allowing developers to choose the programming language they want to practice. These types of pages are used by large companies such as IBM to evaluate developers. Among these pages we can mention [4]:

1. LeetCode is an online platform that focuses on improving programming skills and preparing individuals for technical interviews.
2. CodeSignal is a platform that conducts assessments to facilitate the technical hiring process, from the initial evaluation to the interview.
3. HackerRank supports a variety of programming languages, allowing users to practice and demonstrate their skills in multiple languages.

The aforementioned web pages are used for job interviews and to improve the logic of junior programmers, these pages use similar logic problems so that when developers arrive to an interview, they are better prepared.

B. Programming-focused video game

In this section three video games are presented, which contains a set of puzzles to be programmed to advance thorough a set of levels. Three of them aims to evaluate the developers' knowledge and help them improve their logic.

1. CodeCombat is a website that offers a programming game to learn how to code. In this game, you can learn programming languages such as Python, JavaScript, and HTML while solving puzzles and learning to create your own coding games and websites
2. Robocode is a programming environment where the goal is to develop a battle robot to compete against other tanks. These battle robots are programmed in Java. The environment offers a code editor, a debugger, and a compiler, all in one.
3. Screeps is a massive multiplayer online real-time strategy game. The main feature of the game is that it allows players to program the artificial intelligence (AI) of their units using JavaScript. In other words, you control your colony by writing code.

C. Video game focused on teaching good practices in the software development process

Smellware is a game that aims showing distinct types of smells in software code [5] and the best practices to solve them. This proposal aims to help software developers to recognize diverse types of smells and be aware of the effects of these bad practices in software development.

It is based on an educational game called Riskware [6], a game where the board represents the execution of a project in which different risks that affect the progress of the work team may be materialized. In this game, the team members have an initial budget to purchase resources and controls allowing them to avoid the risks that could be materialized during the development of the game.

III. Description of the proposed platform

This section provides technical information regarding the proposed platform.

A. Requirements

The characteristics and requirements considered for this platform are the next listed:

1) A web application.
2) An intuitive and pleasant interface.
3) User registration and authentication system.
4) Tracking and likes system.
5) A publishing system.
6) An educational video game, intuitively playable.
7) The system must have also an API [7] to be able to manipulate information between two different systems, video game and web system.
8) A section to register/edit questions to the administrator users so that this question can be part of the game.
9) And it must have a real time system to display data.

B. Development methodology

Agility is about quickly responding needs and demands to the market and the customer and being able to change direction as the situation requires. Agile methods attempt to maximize the delivery of value to the customer and minimize the risk of creating products that do not or fail, to meet market or customer needs [8].

Due to the specific features of this project, the agile method chosen to carry out this platform was SCRUM [9].

C. Framework

To build the platform we used different frameworks: Laravel for developing the web system; Unity for developing the video game. Besides, we use the MVC pattern for developing the web system [10].

Fig. 1 shows the pattern will help us to organize correctly the code and to communicate with the client.

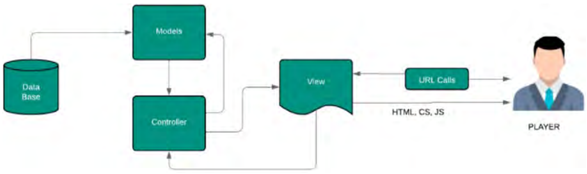

Fig. 1. *Architecture Model, View, Controller.*

Although it is true that the entire web system can be created using the MVC pattern [11], a small portion is developed using the REST pattern,[12].

For the development of the video game a system called FSM [13] was implemented, which stands for Finite State Machine. It allowed us to develop playable and non-playable characters, with a pattern of states. Besides, it helped us to have a more adequate organization of the platform and to reuse code no matter which sprit or character uses it. This way allowed achieving the possibility that admin users register a question on the page as part of the game; and to be able to create our own API to send data with a satisfactory communication between the web system and the video game.

D. Database

For the implementation of the database we use Mysql, so that the database was planned and developed to be relational. The system database has of the following components:

1) Users, its function is to store all user data, it has: unique id, name, email, password, image and username.
2) Publications, to store all the publications that a user can make, composed of: Unique id, title, description, image, and user id.
3) Followers, to store the records of users that follow each other, the table is composed of unique id, user id, and follower id.
4) Likes, to store the likes records that a publication has, this record is made up of unique id, user id, and publication id.
5) Comments, to store the comments that a publication can have, the table has a unique id, user id, publication id, and comment.
6) Levels, which main function is to store the information that a user can have in at a specific moment of finishing a level, this same information will be used to graph the progress of each player. The record is composed of unique id, user id, level number, time, errors, deaths, and extra points.

E. Communication

In order to solve an obstacle that was presented at the time of development of the platform between the web system and the video game, we decided to use the REST framework to create our own API to manipulate the information between systems.

F. Tests

The system starts with the web system interface, where the user authentication is required, done by email and password. It should be noted that the other views are protected, which means that if a person tries to enter the game without going through the start section, it will not allow that person to play.

In the statistics data interface is where the player can see black graphs of the scores, deaths and errors that the player has made, the average time to pass the level, as well as the best time recorded. Additionally, it provides a list of the best scores of all active players [12].

G. Software quality improvement activities

The following section aims to justify how the creation of this video game will help players to implement good software engineering practices in the future. The case proposed in this paper encourages the use of best development practices in coding and testing. Then, the use of these video game can help to meet requirements and deliver value to users [14].

1) *Problem decomposition:* the video game is designed to present complex puzzles and challenges that require proper problem decomposition. Players must break challenges into smaller, more manageable tasks, which encourages the practice

of breaking down complex problems into simpler and more reachable goals. This skill is critical in software engineering, as it allows for the development of more structured and maintainable solutions.

2) *Modularity and code reuse:* as players progress through the game, they face increasingly difficult challenges that require more complex solutions. Players learn to break their code into independent, reusable modules, which facilitates future problem solving and promotes greater efficiency in software development.

3) *Testing and debugging:* the video game aims to develop the ability to test and debug players' code by facing errors and bugs in the execution of their solutions, so players gain experience in debugging techniques and learn the importance of testing. By this way, the video game introduces the idea that bug detection and correction are an integral part of the software development process, thus encouraging good practices in terms of code quality and robustness.

4) *Improved code quality:* code review allows identifying and

correcting bugs, inefficiencies and bad practices in the code developed by the players. By submitting their code to peer review or subject matter experts, players can receive valuable feedback on how to improve their coding style, optimize algorithms and apply better programming practices. This leads to improved code quality and the ability of players to develop more efficient and robust solutions.

5) *Code maintenance:* in the video game context, players can learn and practice code maintenance techniques, such as refactoring and bug fixing, as they develop solutions to puzzles. Regular practice of code maintenance helps players understand the importance of readability and modularity, leading to cleaner, easier to understand and maintain code. Also, by fostering the habit of maintaining code in good condition, players develop a quality-oriented mindset and continuous improvement.

The main view of the development platform in Unity [13], which contains all the graphical part of the game, as well as functionalities and animations. This platform has a function called "Platform-dependent compilation" and consists of a way to compile and execute the code to evaluate it on the platform where the video game is presented.

H. Public repository

The platform was carried out under the Creative Commons license, which means we share the base code and the link to try the game.

- https://github.com/InventoresDIVTIC/DevGAME-Unity
- https://devgame.inventores.org/

Justification of systems architecture and programming

To build the 2D design [15], animations, as well as the functionalities of the video game were developed with the Unity engine [16], using the C# programming language for the development of the scripts focused on object-oriented design [17]. For the interactive visualization of the video game, it was necessary to create a web page, exporting it in WebGL format [18], [19] and implemented in the JavaScript programming language to render graphics and thus independently represent them as interactive graphics in the web browser [20].

The web page, created with the Laravel [22] framework, uses the PHP programming language, allowing us to create it in a simpler and more presentable way. Likewise, the CSS framework called Tailwind [23] was used in the design part for its presentation.

For the database management system, we used MySQL, where all the information related to the web page and the users is stored.

Justification of Intelligent Systems

As mentioned above, we decided to make a more complete implementation for the mechanics performed by the enemies, in order to make the interaction with the users more dynamic. To achieve this, we developed different implementations focusing on the operation of an intelligent agent developed with state machines, in addition; we implemented a decision tree and a search algorithm called "A* Pathfinding"[12] when deciding the shortest route in the direction of the player.

In the implementation of the A* Pathfinding algorithm to find the shortest path to the player, we use the Euclidean distance as a heuristic. The Euclidean distance is calculated in a two-dimensional (x, y) plane and represents the straight distance between two points. In our case, these points are the current position of the enemy and the position of the player.

The A* Pathfinding algorithm uses this Euclidean distance as an estimation of how far it is to the target, which allows us to select the most favorable nodes in the optimal pathfinding. By combining the Euclidean distance with the distance traveled so far, A* Pathfinding can make informed decisions about which nodes to explore next, and thus avoiding obstacles in the environment.

Node selection is performed based on the value of an evaluation function called "f(n)". For each candidate node, the value of "f(n)" is calculated by summing the distance traveled from the starting point to the current node (denoted "g(n)") and the estimated distance from the current node to the target (denoted " h(n)").

g(n): It is the cumulative cost from the initial node to the current node. In other words, it represents the distance traveled from the start to the current node on the route currently under consideration.

- h(n): This is the heuristic function that estimates the distance from the current node to the target. In this case, the Euclidean distance is used to calculate it. The specific formula is: "h(n) = sqrt((n.x - goal.x)^2 + (n.y - goal.y)^2). Where "n.x" and "n.y" represent the (x, y) coordinates of the current node, and "goal.x" and "goal.y" represent the (x, y) coordinates of the goal (player's position).
- "f(n)": It is the total value of the evaluation function and is calculated by adding the accumulated cost "g(n)" and the estimated distance "h(n)". Therefore, "f(n) = g(n) + h(n)". The value of "f(n)" represents an estimate of the total cost from the start to the target via the current node.
- The A* Pathfinding algorithm selects the node with the lowest value of "f(n)" at each step, which means that the nodes that seem most promising in terms of finding an optimal route are explored first. This combination of distance traveled and estimated distance helps guide the enemy to the player in the most efficient way possible, avoiding obstacles and finding the shortest route.

Justification of Distributed Systems

A programming-related question and answer system was implemented on the web site with the ability to manage and modify questions. There are two user roles: administrator and common user. The administrator can add, delete and edit questions, which will then be displayed in the video game. Different answer options can be added to increase the diversity of the quizzes. Changes made by the administrator are instantly reflected on the website. The video game displays random questions and switches questions when the user answers incorrectly, applying a penalty. This allows feedback and scores to be obtained. After answering each question, users can see their performance on the website, indicating whether it was correct or incorrect.

As result of developing the platform: 1) a presentable, intuitive, and interactive web system was developed to be as accessible to the user without the need to install a third-party program; 2) a web system for enabling an authentication process was developed, with the purpose of having security and a way for users to communicate through a forum; 3) we achieved a successful implementation of microservices for the manipulation of information between the web page and the video game, and 4) an educational video game was developed to measure the level of programming knowledge of a user and interactively encourage good software development practices.

I. Guees-18 methodology

Integration and validation of the Game User Experience Satisfaction Scale (GUESS-18) was performed within “Logic Driven Learning,” an educational video game designed to improve software engineering skills. Our methodology encompasses participant selection, gameplay, data collection and analysis to evaluate the impact of the game on learning outcomes and user satisfaction which are described below.

A. Participant Selection

For this study, the participant selection process was meticulously designed to gather a representative sample from a specific academic setting. We recruited a total of 111 students from the University of Guadalajara, specifically from the University Center of Exact Sciences and Engineering. This cohort was chosen based on their background in programming, ensuring that all participants had a fundamental understanding of software engineering principles.

In terms of demographic criteria, the age range of the participants was set between 18 and 28 years. This age bracket was chosen to represent a typical undergraduate and early graduate student population, which is reflective of the primary user base for the "Logic-Driven Learning" game. The inclusion of participants who were already acquainted with programming was crucial. It ensured that the study's findings would be relevant and applicable to students actively engaging with software engineering concepts, thus providing more meaningful insights into the game's effectiveness as an educational tool.

By selecting participants from the University Center of Exact Sciences and Engineering at the University of Guadalajara, the study aimed to maintain a high level of consistency in the educational background of the participants. This consistency was pivotal in assessing the impact of the game on enhancing software engineering skills and in evaluating the user experience through the GUESS-18 survey with a homogeneous group, ensuring that the results were both relevant and applicable to the field of software engineering education.

B. Pre-Game Setup

Before the commencement of the gameplay, a critical stage of preparation, termed the Pre-Game Setup, was diligently conducted to establish a baseline and ensure all participants were adequately oriented. This phase was essential for aligning the participants' expectations and understanding of the study's purpose and the game's objectives.

Orientation Session: An integral component of this phase was the Orientation Session. Here, participants were introduced to "Logic-Driven Learning," the educational video game at the heart of the study. This session was not merely a walkthrough of the game mechanics but also an informative briefing on the research objectives. Participants were informed about the importance of their feedback for the study and how it would contribute to enhancing educational tools in software engineering. The session aimed to foster a sense of engagement and responsibility among the participants, encouraging them to be attentive and reflective during their gameplay experience.

Baseline Data Collection: Alongside the orientation, a Pre-Game Survey was administered. The purpose of this survey was to gather crucial baseline data regarding the participants' prior knowledge and experience in both software engineering and gaming. This data would later serve as a comparative measure to assess the impact of the game on the participants' learning and user experience. By understanding their initial skill levels and attitudes towards gaming and software engineering, the research team could more accurately evaluate the game’s effectiveness and the participants' progress.

The Pre-Game Setup phase was meticulously planned to ensure that all participants started the game with a clear understanding of what to expect and how their involvement was pivotal to the research. It set the stage for a controlled and informed study environment, crucial for the validity and reliability of the research outcomes.

Integrating the GUESS-18 survey into "Logic Driven Learning" required a thoughtful and personalized approach, ensuring that the survey resonated with the specific context of the game while maintaining its primary purpose of evaluating the user experience.

The process for Integrating the survey into the game was approached with the player experience in mind. We strategically placed the GUESS-18 survey prompts at points in the game where they would be most meaningful and least intrusive. For example, after completing a significant level or after a particularly challenging software engineering problem, the survey was presented to players. This moment was chosen to capture immediate, thoughtful feedback while maintaining a seamless gaming experience.

For final testing before full deployment, this version of GUESS-18 underwent a series of tests with a small group of 4 to 5 players. Their feedback was used to further refine the survey wording and integration points within the game, ensuring clarity and relevance.

Through this process the GUESS-18 survey was effectively integrated into "Logic Driven Learning," prepared to capture valuable information about players' learning experiences and satisfaction levels. This integration was instrumental in evaluating the effectiveness of the game as an educational tool in software engineering and in understanding the nuances of the user experience in an educational game context.

C. Data Collection

The Data Collection phase of the study, integral to the evaluation of "Logic-Driven Learning," was meticulously structured to capture a comprehensive understanding of the players' experiences using the GUESS-18 survey, supplemented by additional demographic and experience-related questions.

Implementation of GUESS-18 via Google Forms: To facilitate the collection of survey data, we utilized Google Forms, a reliable and user-friendly platform. This approach not only streamlined the data collection process but also provided a familiar interface for participants, potentially increasing response rates and accuracy. The Google Forms survey embedded the 18 questions from the GUESS-18, carefully adapted and integrated as previously outlined.

Additional Questions for Contextual Insight: Alongside the standard GUESS-18 questionnaire, four additional questions were incorporated into the survey. These questions were designed to gather demographic data and information about the participants' programming experience, crucial for contextualizing their feedback on the game. The added questions were:

- Age: To understand the age distribution of our participant group and to see if there were any age-related trends in the feedback.

- Sex: To capture gender-based perspectives, if any, on the gaming and learning experience.

- Programming Experience: To ascertain whether participants had prior experience in programming, which could influence their interaction with the game and their perception of its educational value.

- Self-Assessment of Programming Skills: Participants were asked to rate their own programming skills. This self-assessment helped in correlating perceived skill levels with user experience and satisfaction, providing insights into how the game benefits learners at different skill levels.

Deployment and Response Monitoring: The survey was deployed at specific intervals during gameplay, as well as at the conclusion of the gaming session. The timing was strategically chosen to capture feedback that was both immediate and reflective of the overall experience. Continuous monitoring of survey responses was carried out to ensure a high response rate and to address any technical issues promptly.

Through this comprehensive data collection approach, employing Google Forms for ease of use and adding crucial demographic and experience-related questions, the study aimed to gather rich, multi-dimensional data. This data was pivotal in assessing the educational impact and user experience of "Logic-Driven Learning," thereby contributing to the broader understanding of user engagement and satisfaction in educational gaming environments.

IV. Results and discussions

The Fig. 2 show bar's graph that represents the age distribution of 111 participants who completed a Google Forms survey for the usability study of the "Logic-Driven Learning" game. Ages range from 17 to 40 years, with the majority of responses concentrated in the 25-year-old group, which has 24 participants (21.6%). The next highest concentrations of responses are at ages 23 and 21 years, with 21 (18.9%) and 14 (12.6%) participants, respectively. The distribution indicates that the predominant group of players providing feedback consists of young adults in the early stages of university or postgraduate education.

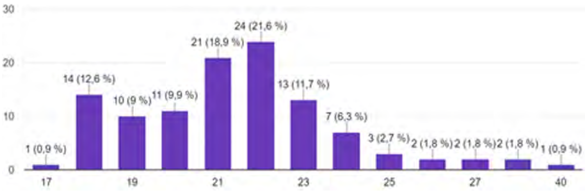

Fig. 2 *Graph of number of respondents and their age.*

The Fig. 3 displays a pie chart reflecting responses to the question "Do you have programming experience?" from the 111 participants who completed the Google Forms survey for the usability study of the "Logic-Driven Learning" game. The chart shows a significant majority, 93.7%, indicating "Yes" (represented in blue), suggesting they have programming experience. A smaller wedge, 6.3% (depicted in red), represents the participants who answered "No," indicating they do not have programming experience. This data highlights that the vast majority of respondents are familiar with programming, providing context for the survey responses within the domain of software engineering education.

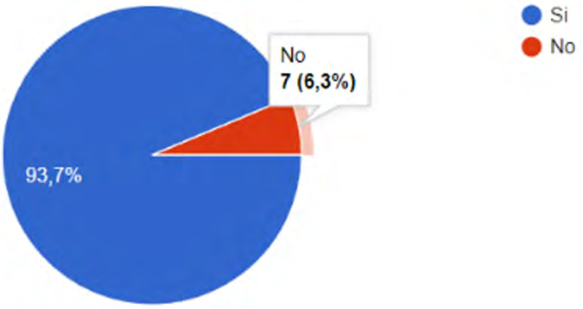

Fig. 3 *Pie chart reflecting responses to the question "Do you have programming experience?"*

Fig. 4 is a bar graph summarizing responses to a usability question from the GUESS-18 survey about the game "Logic-Based Learning." Most participants found the game's controls easy to use, with 78 of 111 indicating they agreed or strongly agreed that the controls are straightforward and simple. A minority of 18 participants disagreed or strongly disagreed with the statement, while 15 participants were neutral.

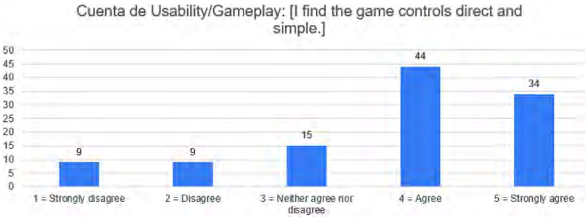

Fig. 4 *Graph of the game controls direct and simple.*

Fig. 5 shows responses to the GUESS-18 survey question about the usability of the game interface, specifically whether players find the interface easy to navigate. The majority of participants responded positively: 45 agree (4 on scale) and 40 strongly agree (5 on scale), suggesting that they find the interface easy to use. A minority of respondents, 7 strongly disagree (1 on scale) and 6 disagree (2 on scale), did not find the interface easy to navigate. Thirteen participants are neutral (3 on the scale). The results indicate that the majority of players had a favorable experience with the game interface.

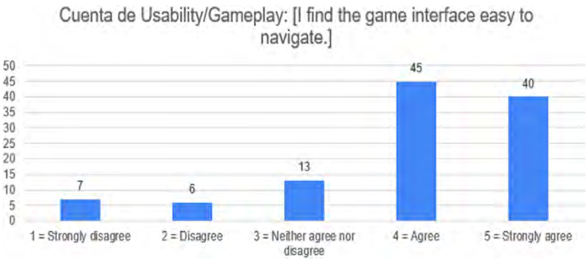

Fig. 5 *Graph of the game interface easy to navigate.*

Fig. 6 shows responses about the game's narrative, specifically whether players are captivated by the story from the beginning. The most common response is neutral: 43 participants selected "Neither agree nor disagree" (3 on the scale). This is followed by 25 participants who "strongly disagree" (1 on the scale) and 21 who "disagree" (2 on the scale), indicating a lack of engagement with the story. At the other extreme, 16 agree (4 on the scale) and 6 strongly agree (5 on the scale), which shows a certain level of commitment. These results suggest that while some players are interested in the story, a notable portion of them are not enthralled from the start.

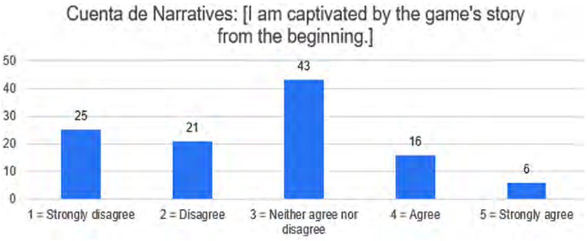

Fig. 6 *Graph of captivated by the game's story from the beginning.*

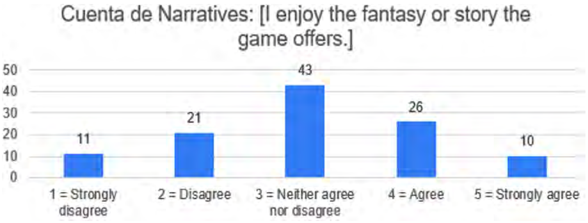

Fig. 7 *Graph of the fantasy or story the game offers.*

Fig. 7 presents responses to the narrative engagement question from the GUESS-18 survey: "I enjoy the fantasy or story the game offers." The majority of participants, 43, are neutral, neither agreeing nor disagreeing. Twenty-six participants agree with the statement and 10 strongly agree, indicating that they enjoy the story of the game. In contrast, 21 disagree and 11 strongly disagree, suggesting that they do not enjoy the narrative as much. This distribution implies a varied reception of the game's story among players.

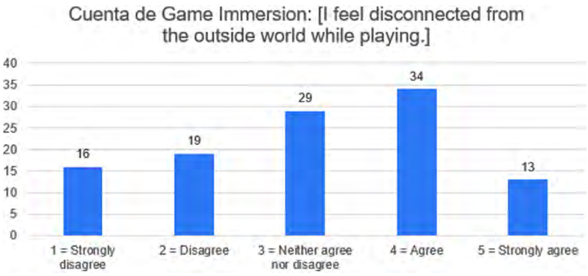

Fig. 8 *Graph of the outside world while playing.*

Fig. 8 is a graph that describes how players feel disconnected from the outside world while playing. The results show a distribution across the spectrum: 16 participants strongly disagree with feeling disconnected, 19 disagree, 29 neither agree nor disagree, 34 agree, and 13 strongly agree. Most participants feel some level of immersion, with a combined total of 47 of 111 indicating that they agree or strongly agree with the statement, suggesting that the game has a substantial immersive effect on these players.

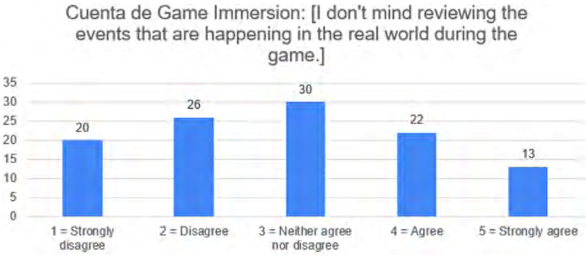

Fig. 9 *Graph of the events that are happening in the real world during the game.*

Fig. 9 shows responses to a GUESS-18 survey question on game immersion: "I don't mind reviewing events that happen in the real world during the game." Player opinions appear to vary: 20 strongly disagree with being indifferent to real-world events while playing, suggesting high immersion, while 26 disagree and 30 neither agree nor disagree. . Meanwhile, 22 agree and 13 strongly agree, indicating that they are not as immersed and do not mind being connected to the real world during the game. The data suggest a balanced distribution of immersion levels among participants.

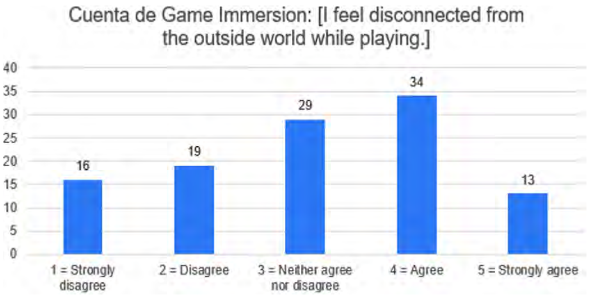

Fig. 10 *Graph of the game is fun.*

Fig. 10 is the responses to the enjoyment question from the GUESS-18 survey: “I think the game is fun.” A considerable number of participants, 40, agree that the game is fun, while 23 strongly agree, indicating a high level of enjoyment. On the other hand, 7 participants totally disagree and 13 disagree that the game is fun. Twenty-eight participants are neutral, neither agree nor disagree. Overall, most responses lean towards the game being perceived as fun.

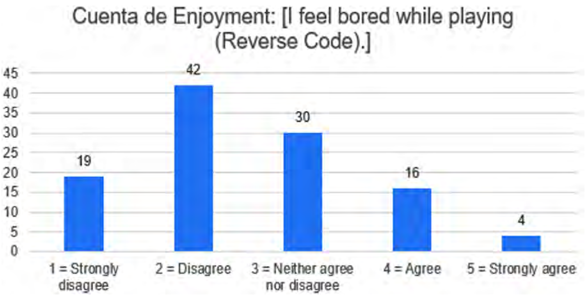

Fig. 11 *Graph of the bored while playing.*

Fig. 11 indicates that 42 participants disagree and 19 strongly disagree with feeling bored, suggesting that they find the game engaging. Thirty participants adopt a neutral stance, while a minority of 16 agree and 4 strongly agree with feeling bored, indicating a lower degree of enjoyment. The majority sentiment leans towards the game not being boring for the players.

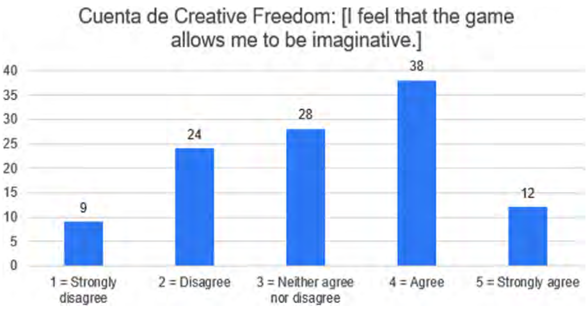

Fig. 12 *Graph of the game allows me to be imaginative.*

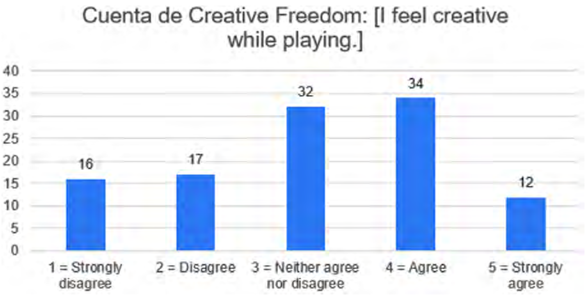

Fig. 13 *Graph of the creative while playing.*

The Fig. shows a distribution in which most players agree (34 responses) or strongly agree (12 responses) that they feel creative during the game, indicating a sense of creative engagement. A moderate number of participants are neutral (32 responses), while fewer participants disagree (17 responses) or strongly disagree (16 responses) with the statement, suggesting that they feel less creative freedom while playing. Overall, the answers lean towards gameplay, providing a sense of creative freedom to players.

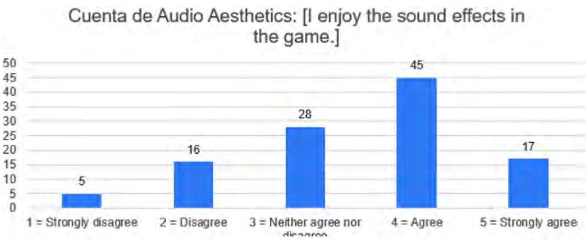

Fig. 14 *Graph of the sound effects in the game.*

Fig. 14 shows the results of a GUESS-18 survey question on audio aesthetics: “I enjoy the game's sound effects.” The majority of participants, 45, say they enjoy the sound effects, while 17 strongly agree, indicating a positive reception of the game's audio. A moderate number, 28, are neutral and neither agree nor disagree. Fewer respondents, 16, disagree, and an even smaller group, 5, strongly disagree with enjoying the sound effects. This suggests that most players are satisfied with the game's audio experience.

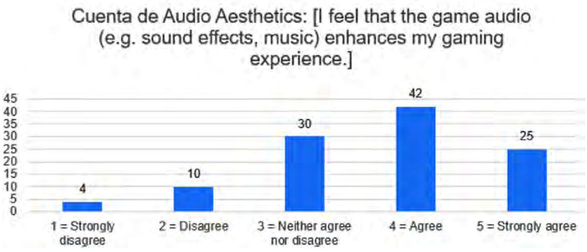

Fig. 15 *Graph of the game audio.*

Fig. 15 shows a bar graph from the GUESS-18 survey, illustrating responses to the statement "Audio Aesthetics: [I feel that game audio (e.g., sound effects, music) enhances my gaming experience.]". Most participants agree (42) or strongly agree (25) that audio contributes positively to their gaming experience. Thirty participants are neutral, while a smaller number disagree (10) or strongly disagree (4). The data suggests that most gamers generally perceive in-game audio as a factor that enhances the gaming experience.

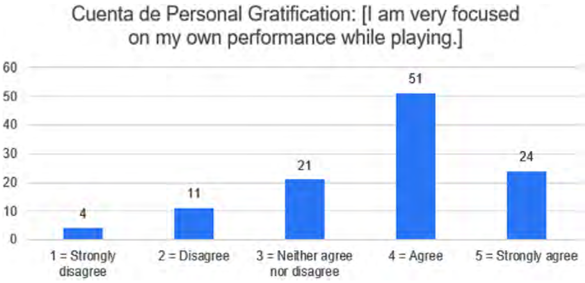

Fig. 16 *Graph of the performance while playing.*

Fig. 16 shows responses to an item from the GUESS-18 survey on Personal Gratification: "I am very focused on my own performance while playing." The largest group of respondents, 51, agree with the statement, suggesting a high level of self-focus on performance. Twenty-four strongly agree, further emphasizing this approach. A smaller number of participants, 21, are neutral, while 11 disagree and only 4 strongly disagree, indicating that they are less concerned about their performance. This indicates that most players are attentive to their performance during the game.

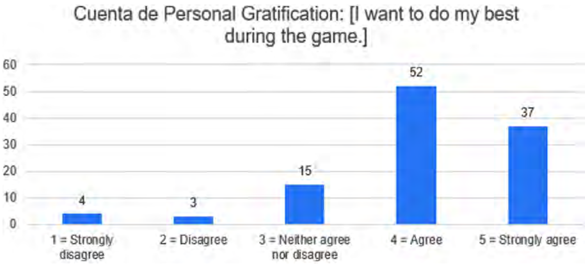

Fig. 17 *Graph of the Personal Gratification*

Fig. 17 shows responses to the Personal Gratification survey question: "I want to do the best I can during the game." A significant majority of players agree (52 responses) or strongly agree (37 responses) with the desire to perform well, suggesting a high level of motivation and personal investment in the game. Only a small number of participants disagree (4 responses) or strongly disagree (3 responses), and 15 participants are neutral on the matter. This shows that most players are committed to excelling in the game.

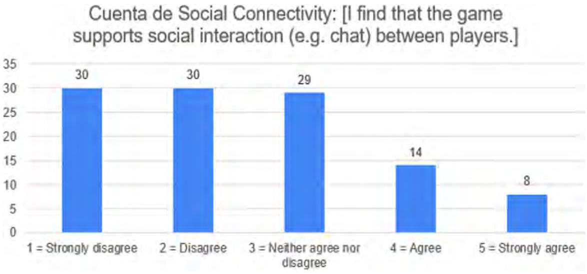

Fig. 18 *Graph of the the game supports social interaction.*

Fig. 18 shows responses to the question about social connectivity: "I think the game supports social interaction (e.g. chat) between players." The most common responses are neutral, with 30 participants neither agreeing nor disagreeing. Both those who agree and those who disagree have responses equal to 30 each, indicating a division of opinion on the social characteristics of the game. A minority of participants, 14, agree and 8 strongly agree that the game supports social interaction, suggesting that while some find the social aspects adequate, others do not feel that the game facilitates interaction.

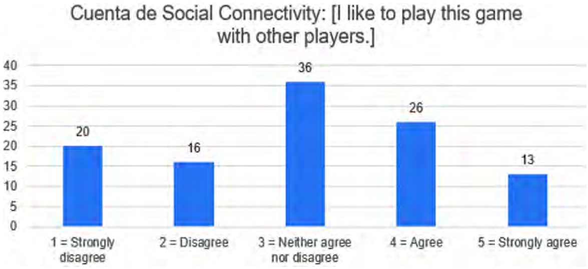

Fig. 19 *Graph of the Social Connectivity.*

Fig. 19 shows that the largest group of respondents, 36, are neutral. Twenty respondents strongly disagree with the preference of playing with others, while 16 disagree. In contrast, 26 agree and 13 strongly agree that they enjoy the game more when playing with others, indicating a variety of preferences for social play within the game.

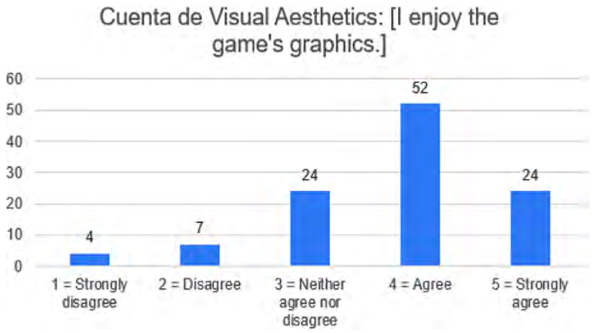

Fig. 20 *Graph of the Visual Aesthetics.*

Fig. 20 shows responses to the visual aesthetics question from the GUESS-18 survey: "I enjoy the game's graphics." The majority of participants, 52, agree that they enjoy the graphics, while 24 strongly agree, indicating a positive response to the game's visual design. A smaller group of 24 respondents is neutral, neither agreeing nor disagreeing. Very few respondents disagree (7) or strongly disagree (4), suggesting that the graphics are generally well received among gamers.

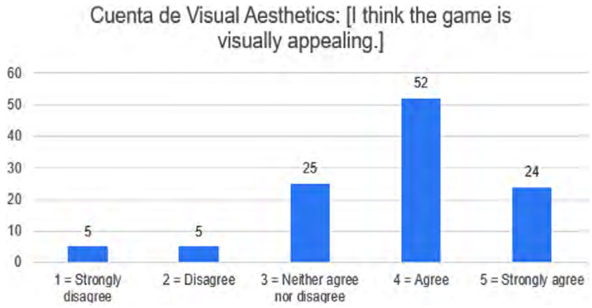

Fig. 21 *Graph of the game is visually appealing.*

Fig. 21 shows the most common response is agree: 52 participants selected “Agree” and an additional 24 chose “Strongly Agree,” indicating a favorable opinion of the game's visual design. A smaller number of participants, 25, are neutral, while very few respondents disagree (5) or strongly disagree (5) with the visual appeal of the game. This suggests that most players find the game's visuals attractive.

V. Conclusions and future work

After reviewing the 18 graphical representations of the GUESS-18 assessment for "Logic-Based Learning: An Interactive Video Game to Encourage Software Engineering," several nuanced conclusions are drawn that offer a complete picture of user experience and educational effectiveness. of the game:

- Usability and control mechanics:
  The data overwhelmingly indicates that the game's control mechanics are easy to use, and most players find them straightforward and simple. This suggests that the game developers have successfully implemented an intuitive interface, which is crucial to maintaining player engagement and facilitating the learning process within the software engineering context of the game.

- Game Immersion:
  There seems to be a wide range of experiences related to immersion, and a notable portion of players feel disconnected from the outside world, a hallmark of a compelling gaming experience. However, the split in responses also points to a segment of players who remain less absorbed. This dichotomy suggests there is room to refine game design to consistently improve immersion across the entire player base.

- Social Connectivity:
  Responses reflect ambivalence regarding the game's facilitation of social interaction. While some players appreciate the social elements, a significant portion remain neutral or dissatisfied. This indicates potential opportunities to deepen the social features of the game, possibly improving communication tools or collaborative challenges within the game.

- Enjoyment and commitment:
  The game appears to score high on enjoyment, with many players expressing that they find the game fun and are willing to give their best while playing, indicating high levels of engagement and motivation. This positive feedback is crucial as it underlines the potential of gaming as an effective educational tool, harnessing enjoyment to improve learning outcomes.

- Personal gratification:
  Responses show that players generally focus on their performance, suggesting that the game successfully encourages players to invest in skill development and mastery, which are key aspects of educational games.

- Creative freedom:
  A considerable number of players feel a sense of creative freedom while playing, indicating that the game likely offers elements that encourage creative problem solving, a valuable skill in software engineering.

- Audiovisual and Visual Aesthetics:
  The visual and audio elements of the game receive positive feedback and most players enjoy the graphical and audio experience, which contributes positively to their overall gaming experience. High-quality audiovisuals are often associated with higher levels of user satisfaction and can play an important role in facilitating an immersive learning environment.

In summary, “logic-based learning” appears to be a well-received educational tool that stands out for its usability, player enjoyment, and aesthetic appeal. However, there are areas for improvement in immersion and social connectivity. The game seems to have struck a chord with its target audience in terms of providing a compelling educational experience that is both enjoyable and conducive to learning. In the future, these insights could guide iterative improvements, improving the game's strengths and addressing its weaknesses to better serve its educational purpose.

Building on the insights gained from the GUESS-18 evaluation, future work on "Logic-Driven Learning: An Interactive Video Game to Foster Software Engineering" could focus on enhancing player immersion. A detailed analysis of the moments where players felt most and least immersed could inform the development of new game levels or features. Immersion can be deepened through richer storytelling, dynamic challenges that adapt to the player's skill level, and more interactive game mechanics that closely simulate real-world software engineering problems. Additionally, leveraging virtual reality (VR) or augmented reality (AR) could offer a more immersive experience, potentially increasing the user's sense of presence within the game.

Social connectivity also presents an avenue for expansion. Given the mixed responses about the game's current social features, future iterations could introduce more collaborative projects or pair programming challenges that require communication and cooperation between players. Community-driven features like leaderboards, forums for sharing solutions, and in-game rewards for teamwork could foster a stronger sense of community. Moreover, integrating social media functionalities could allow players to share achievements and insights, thereby extending the game's reach and impact.

Lastly, there is room for further tailoring the game to the educational objectives of software engineering. Continuous user feedback loops could be established to refine the game's content and ensure that it remains up-to-date with the latest industry practices and technologies. Collaborations with educational institutions and software engineering professionals could provide real-world applicability to the game's scenarios. Research into the game's long-term impact on players' learning outcomes, career trajectories, and retention of software engineering concepts could validate and shape the game's role as an effective educational tool.

VI. Referencias

[1] B. Whitfield. "What Is a Game Developer? How to Become One, Salary, Skills. | Built In". built in. https://builtin.com/learn/careers/game-developer (Accessed July 12, 2023)

[2] D. Bonilla Carranza, A. Peña Pérez Negrón y M. Muñoz, “Logic-Driven Learning: An Interactive Video Game to Foster Software Engineering”, Encuentro Nac. Comput. 2023, 2023.

[3] Keebler, J. R., Shelstad, W. J., Google,

D. C. S., Chaparro, B. S., & Phan, M. H. (2020). Validation of the GUESS-18: A Short Version of the Game User Experience Satisfaction Scale (GUESS). http://uxpajournal.org.

[4] J. Carrillo, “Los 10 Sitios Web Más Populares para Desafíos de Programación [Actualizado para 2020]”, freecodecamp.org, 01-dic-2020. [Online]. Available at: https://www.freecodecamp.org/espanol/news/los-10-sitio s-web-mas-populares-para-desafios-de-programacion-actualizado-para-2020/. [Accessed: 31-ene-2023].

[5] C. M. Zapata, M. C. Gomez y J. C. Hernandez. "SMELLWARE: un juego para la enseÃ±anza de buenas prÃ¡cticas en el proceso de desarrollo de oftware". SciELO - Scientific electronic library online. https://www.scielo.cl/scielo.php?script=sci_ext&amp;pid=S0718-33052020000400645 (accedido el 12 de julio de 2023).

[6] C. Zapata Jaramillo, M. Gomez Álvarez and G. González Calderón. “Riskware: A Game for Teaching Software Project Risk Management”. Developments in Business Simulation and Experiential Learning. Vol. 40. 2013.

[7] "¿Qué es una API REST? | IBM". IBM - Deutschland | IBM. https://www.ibm.com/mx-es/topics/rest-apis (Accessed July 12, 2023).

[8] M. Alqudah and R. Razali, "A Review of Scaling Agile Methods in Large Software Development," International Journal on Advanced Science, Engineering and Information Technology, vol. 6, no. 6, pp. 1-10, Dec. 2016. [Online]. Available: https://dx.doi.org/10.18517/IJASEIT.6.6.1374

[9] "Scrum: qué es, cómo funciona y cómo empezar | Atlassian". Atlassian. https://www.atlassian.com/es/agile/scrum (Accessed July 12, 2023).

[10] "Información general de ASP.NET MVC". Microsoft Learn: Build skills that open doors in your career. https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview (Accessed July 12, 2023).

[11] Introduction to A*". Loading... https://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html (accessed July 12, 2023).

[12] “Qué es MVC”, Desarrolloweb.com, 02-ene-2014. [Online]. Available atn: https://desarrolloweb.com/articulos/que-es-mvc.html. [Consultado: 2023].

[13] reeCodeCamp.org. "Finite State Machine Explained". freeCodeCamp.org. https://www.freecodecamp.org/news/finite-state-machines/ (accedido el 12 de julio de 2023).

[14] Download Archive y Beta Program, “ProgressBar Pack”, Unity.com. [Online]. Available at: https://assetstore.unity.com/packages/tools/gui/progressb ar-pack-120981. : 2023].

[15] Download Archive y Beta Program, “Medieval pixel art asset FREE”, Unity.com. [Online]. Available at: https://assetstore.unity.com/packages/2d/environments/m edieval-pixel-art-asset-free-130131. [Accessed: 2023].

[16] E. Román. "Mejores Prácticas de Desarrollo de Software". Centro de Mejora - El blog de Innevo. https://blog.innevo.com/mejores-practicas-desarrollo-software (Accessed July 12, 2023).

[17] “Unity”, Microsoft. [Online]. Available at: https://dotnet.microsoft.com/es-es/apps/games/unity. [Accessed: 31-ene-2023].

[18] M. Pêcheux, “How to create a simple behaviour tree in Unity/C#”, Geek Culture, 02-nov-2021. [Online]. Available at: https://medium.com/geekculture/how-to-create-a-simple-behaviour-tree-in-unity-c-3964c84c060e. [Consultado: 2023].

[19] Unity Technologies, “Empezar con el desarrollo de WebGL”, Unity3d.com. [Online]. Available at: https://docs.unity3d.com/es/2018.4/Manual/webgl-gettingstarted.html. [Accessed: 2023].

[20] “Primeros pasos con WebGL”, Mozilla.org. [Online]. Available at: https://developer.mozilla.org/es/docs/Web/API/WebGL WebGL_ API/Tutorial/Getting_started_with_WebGL. [Accessed: 2023].

[21] Download Archive y Beta Program, “Pixel Adventure 2”, Unity.com. [Online].

Available at: https://assetstore.unity.com/packages/2d/characters/pixel-adventure-2-155418. [Accessed: 2023].

[22] "Installation", Laravel.com. [Online]. Available at: https://laravel.com/docs/9.x. [Consultado: 2023].

[23] "Documentation - Tailwind CSS". Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. https://v2.tailwindcss.com/docs (Accessed July 12, 2023).