RetoñosApp: Work in Progress on a Platform to Support the Teaching of Programming in CS through the Automation and Customization of Learning Processes Guided by Artificial Intelligence

David Mauricio Valoyes-Porras[a], Juan Sebastián Rodríguez-Obregón[b], David Steven Salamanca-Sánchez[c] and Miguel Alfonso Feijóo-García[d]

Program of Systems Engineering, Universidad El Bosque, Bogotá, Colombia

Keywords: User-centered Interaction, User Experience, Artificial Intelligence, Recommendation System, Computer Science Education.

Abstract: Learning difficulties in Computer Science (CS) are a multicausal problem that promotes student dropout in CS undergraduate programs. This results from the students' psychological, emotional, and motivational implications, affecting their academic performance. We present RetoñosApp, as a web-based and user-centered platform supported by Artificial Intelligence (AI) that assists the teaching and learning processes for CS. It fosters the students' autonomous learning, and provides accompaniment and feedback to students during their academic term, and CS instructors on their students' learning processes. This web-based tool uses a Conversational Bot as an autonomous and synchronous virtual tutor, and a Content-based Recommendation System to generate customized reports with "educational routes" to students and instructors, based on their needs. We evaluated this web-based tool, and reported the findings and results, considering its efficiency and effectiveness, based on the participants' interaction. This, in order to answer how the platform supported and complemented the teaching and learning processes of programming in CS, evaluating its potential to be part of the educational methodology of further CS undergraduate courses. Our findings from the pilot study suggest that RetoñosApp effectively provides a friendly user-centered asynchronous assistance and enhancement to learning processes, and frequent feedback on teaching processes.

1 INTRODUCTION

Computer Science (CS) is the discipline that gathers the fundamentals, methods, and theory to support the development of new informatic solutions (Sommerville and Torres, 2011). Consequently, CS education constitutes a considerably high degree of complexity. Programming skills, as basic and fundamental knowledge of CS, present one of the main challenges that professionals in this field often face. These difficulties result from the requirement of teaching-learning processes to be incremental and evolving in CS, based on curricular approaches that gather previous skills and knowledge. This has motivated the CS Education community to evaluate teaching and learning strategies often to face this educational challenge.

[a] https://orcid.org/0000-0003-2893-0261
[b] https://orcid.org/0000-0001-7039-244X
[c] https://orcid.org/0000-0001-9355-3955
[d] https://orcid.org/0000-0001-5648-9966

There are numerous technologies and research with scaffolds on the following topics: (1) interactive platforms as a complement to the educational processes that are synchronous-based, (2) conversational bots as virtual tutors, and (3) recommendation systems for the abstraction of shortcomings and difficulties in educational processes.

In the first place, interactive e-learning tools for synchronous assistance of academic activities encourage the students' participation in constructing their knowledge (Mutiawani et al., 2014). On the other hand, integrating conversational bots as virtual tutors promotes student learning in educational environments mediated by information and communication technologies (ICTs) (Wellnhammer et al., 2020). In addition, implementing recommendation systems in education (Cano and Alarcón, 2021) allows identifying students' learning difficulties to support and provide feedback on educational processes based on their needs (Harris and Kumar, 2018).

Valoyes-Porras, D., Rodríguez-Obregón, J., Salamanca-Sánchez, D. and Feijóo-García, M.
RetoñosApp: Work in Progress on a Platform to Support the Teaching of Programming in CS through the Automation and Customization of Learning Processes Guided by Artificial Intelligence.
DOI: 10.5220/0011538200003323
In *Proceedings of the 6th International Conference on Computer-Human Interaction Research and Applications (CHIRA 2022)*, pages 179-186
ISBN: 978-989-758-609-5; ISSN: 2184-3244

We introduce a web-based tool called RetoñosApp as a friendly and user-centered platform supported by Artificial Intelligence (AI) that assists the teaching and learning processes for CS. It fosters the students' autonomous learning, and provides frequent accompaniment and feedback to students during their academic term, and CS instructors on the overall learning progress of their students. Like LEGO (McNamara et al., 1999), this approach pretends the students to guide their customized "learning route", following significant learning. The Significant Learning Model (Baque-Reyes and Portilla-Faican, 2021) constitutes a constructivist teaching model, where the student is not only a receiver, but the protagonist of their learning process. The constructivist theory (Saldarriaga-Zambrano et al., 2016) asserts that teaching-learning processes should be based on the construction of knowledge from enriching experiences (i.e., individual needs), beyond the simple transmission of concepts or skills.

This paper presents RetoñosApp and the results of our first pilot study using the web-based tool. We piloted the study with first-year undergraduate students (i.e., CS1 and CS2) at Universidad El Bosque, Colombia. We present our findings and results on students' perceptions based on their interaction with the tool, and the impact of our approach on their learning processes. We also describe its limitations, addressing the pros and cons of this preliminary pilot approach.

With this study, we intended to validate the following hypothesis: "Using a teaching-learning platform based on Artificial Intelligence (AI) satisfactorily benefits, supports, and complement the educational processes of programming, providing customized feedback on particular topics to the students, and the overall groups' progress to the instructors". Thus, we ask the following questions: How do we promote a clear customized learning roadmap in CS? How do we make this roadmap to be learner-centered?

Our work contributes to CS Education and Computer-Human Interaction literature, evaluating a web-based approach to support teaching and learning processes in CS in a customizable way. This approach uses the benefits of autonomous, synchronous, and customized educational processes addressing CS concepts and skills.

2 RetoñosApp: DESCRIPTION

We introduce a web-based platform that supports and complements the teaching and learning processes in CS programming courses. It provides a friendly and interactive GUI that allows the user to guide customized "educational routes" through asynchronous assistance and frequent feedback. RetoñosApp features a conversational bot that fulfills the task of an autonomous and synchronous virtual tutor— we implied SAP Conversational AI (Adamopoulou and Moussiades, 2020), which involves Natural Language Processing (Chowdhary, 2020) (i.e., NLP, involving syntactic and semantic processing). Moreover, it features a content-based recommendation system (Pazzani and Billsus, 2007) fed by the information collected by the platform (i.e., user entries) and the conversational bot (e.g., topic-centered questions, personalized user interaction). RetoñosApp provides a customized "educational route" to students, or the overall progress of a group of students of a particular course to the instructors. The customized report is based on the students' particular needs, doubts, and difficulties (see Figure 1).

This approach used techniques and concepts of CS, Data Analytics, and Intelligent Systems. RetoñosApp is structured by three main modules: (1) Administrator, (2) Teaching, and (3) Learning. The first module (i.e., Administrator) presents the management and audit features of the platform (e.g., parameter configuration, user management, incident management, traceability verification, display or exportation of reports). In addition, it features a friendly and user-centered GUI that implies that administrators (i.e., users) do not necessarily have to understand technology to interact with these features. On the other hand, the second module (i.e., Teaching) provides features to instructors (e.g., creation of subjects, creation of topics, creation of class sessions, analysis/exportation of customized "educational routes" of a group of students— strictly related to the recommendation system). Finally, the third module (i.e., Learning) presents to the students the asynchronous work proposed as a support and complement to the teaching process in their programming courses in CS. This third module guides and presents monitoring to the students' customized educational progress— an individual educational roadmap. It also features the interaction with a conversational bot that works as a virtual tutor and a "synchronous" and "autonomous" guide that intends to accompany the student during their learning process on the platform.

Architecturally, RetoñosApp presents a simultaneous, coordinated, and complimentary use of a (1) Clean Architecture (based on a hexagonal architecture) (Martin et al., 2018), (2) a layered architecture— consisting of a combination of traditional layered architecture (Richards, 2015) and a

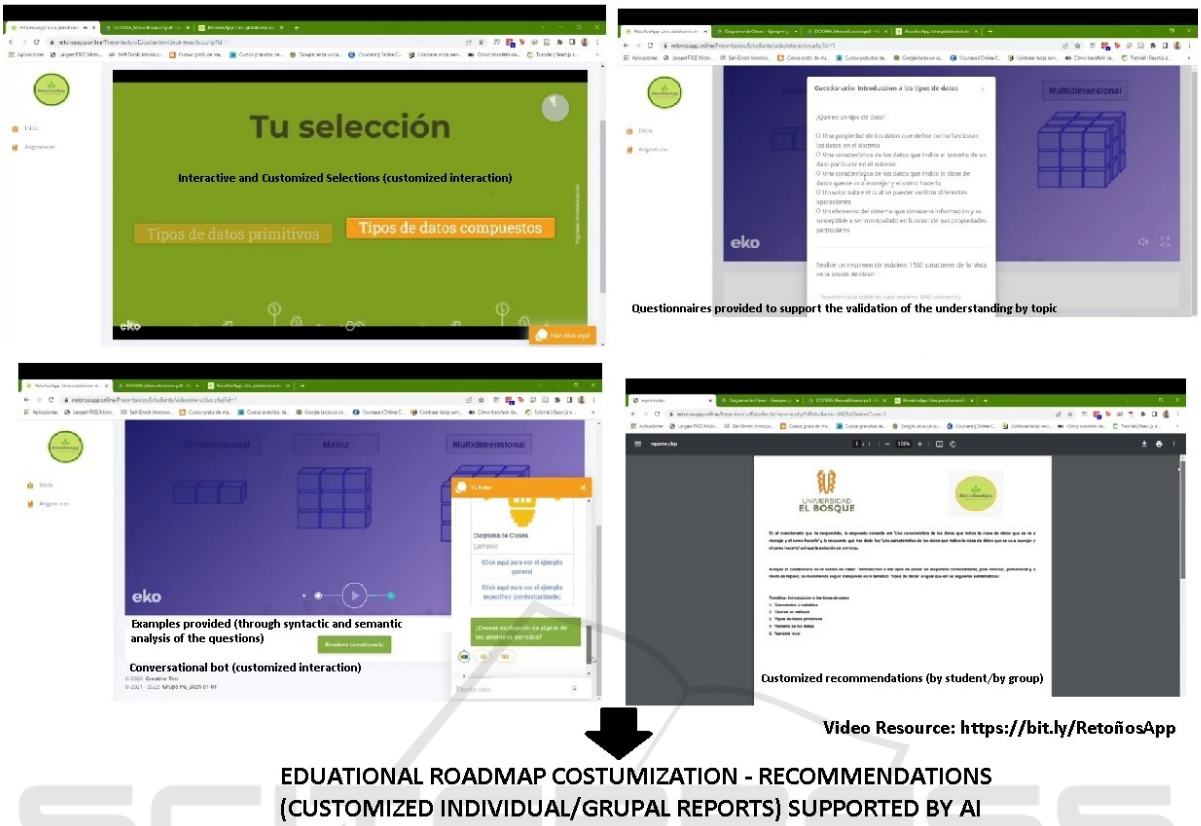

Figure 1: Features/Functionalities of RetoñosApp (preliminary customization features).

Model–View–Controller (MVC) architectural style (Deacon, 2009), and (3) an attribute-based architecture (Klein et al., 1999)— representing the McCall's quality model (Moreno et al., 2010).

The three architectural styles or approaches selected constitute three different perspectives of the application architecture design. Each of the styles used allows presenting and emphasizing a particular aspect of the general architecture of the application, dividing it into: (1) Technological infrastructure and deployment of the artifact, (2) Architectural design based on the internal and specific components of the application, and (3) Determination of the quality model selected as a guide for the quality assurance of the project.

This web-based approach proposes three main differentiator features: (1) Interactivity as an element that gamifies the learning experience, promoting motivation, retention of information, ease in understanding new topics and concepts, and individualizing learning forms. (2) Using conversational bots to perform virtual tutoring tasks in real-time. (3)Using of recommendation systems to provide feedback on the learning processes, consequently allowing adapting or reconciling the teaching processes based on individual and group needs of students (See Figure 2 and Figure 3).

3 DATA ACQUISITION

The participants who gathered the experimentation voluntarily were Junior Undergraduates (i.e., first-year CS students). We expected to have 188 participants in the experimentation. However, we had a total of 136 students (N=136), representing 72.34%: 71.3% (N=97) for CS1, and 28.7% (N=39) for CS2. The participants had an average age between 18 and 19 years old. We considered this a significant population sample to support the claims presented in this experience report.

We sought to answer how the platform supported and complemented the teaching and learning processes of programming in CS, evaluating its potential to be part of the educational methodology of further CS undergraduate courses. Hence, we evaluated the tool's effectiveness, considering the participants' perceptions of their experience using RetoñosApp. We designed a questionnaire that allowed the participants to evaluate their user experience (Hassenzahl

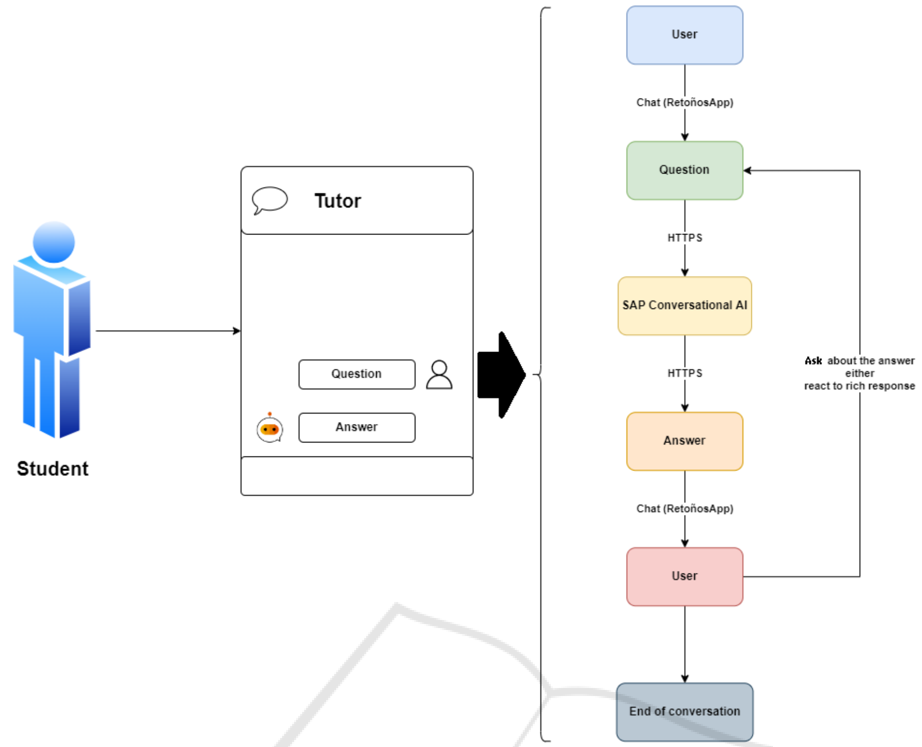

Figure 2: Integration of the Conversational Bot to RetoñosApp.

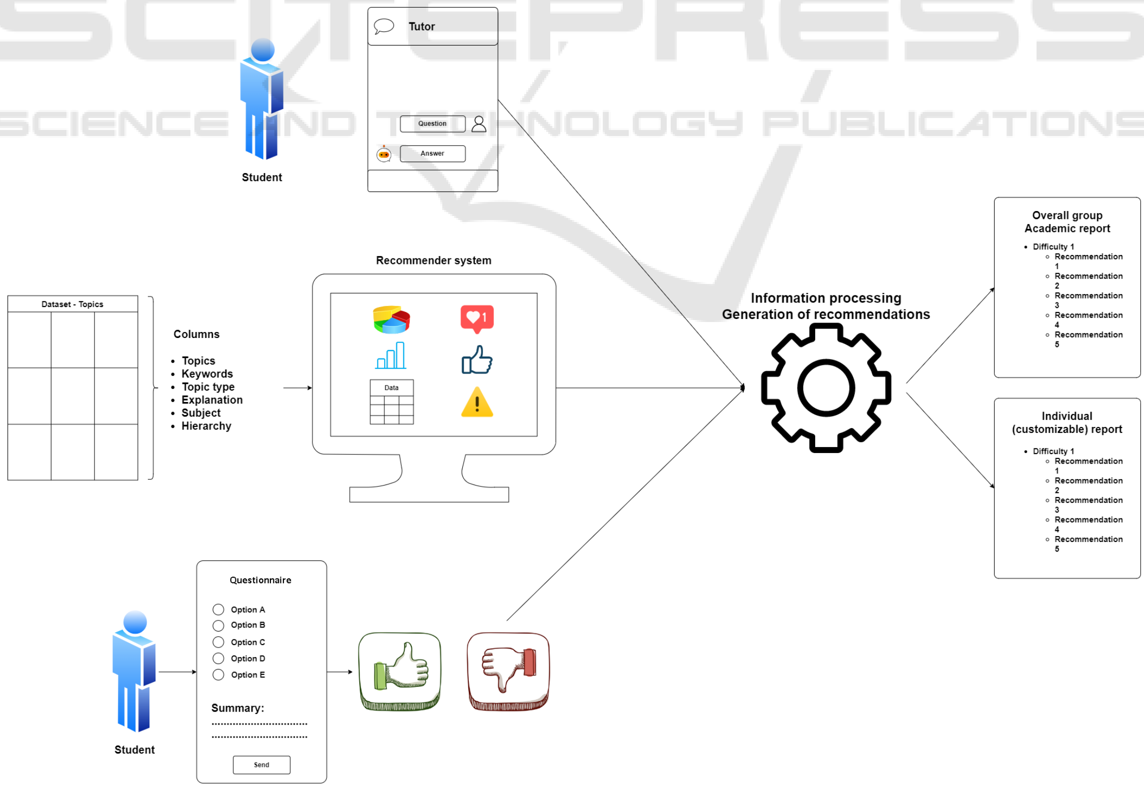

Figure 3: Integration of the Content-based Recommendation System to RetoñosApp.

and Tractinsky, 2006) by interacting with the web-based application. The questionnaire had an initial question of acceptance, guaranteeing the voluntary and uncoerced participation of the students in this pilot study. In addition, it had 5 demographic questions and 22 additional questions focused on measuring the user experience of the target population against the web-based platform (see Table 1).

Perceptions of usability and satisfaction in this study were calculated by applying the following metrics: (1) SUS (i.e., 1-5 System Usability Scales) (Grier et al., 2013), (2) NPS (i.e., 1-10 Net Promoter Scales) (Mandal, 2014), and (3) 1-10 Rating-Scales (Harpe, 2015). For this pilot study, we defined User-Experience as the relation (i.e., synergy) between the participants' perceptions of satisfaction and usability of RetoñosApp (see Table 2).

4 EXPERIMENTAL APPROACH

Each participant followed the experimentation process: (1) Introduction of the experimentation activity based on RetoñosApp. (2) Unsupervised testing phase of the developed application, providing a user manual as a guide for use. This phase lasted between 20 to 30 minutes, depending on the size of each group of CS1 and CS2. (3) Application of a questionnaire to evaluate their user experience on the web-based platform (see Table 1), once they interacted with it.

The user test was uncontrolled (i.e., unsupervised). Apart from the introduction around the experimentation at the beginning of each session, we provided each participant with the system guide of RetoñosApp (i.e., user manual) to test out each of the functionalities freely— they interacted with the platform in the way each participant preferred. Once they interacted with the web-based tool, the participants were given a questionnaire to evaluate their perception of user experience on RetoñosApp through questions regarding Usability (i.e., Functionality, Accessibility, Visual Design, Error Rate, Ease of Learning, Efficiency of Use), and Satisfaction. We asked in this questionnaire (1-5) System Usability Scales (SUS), (1-10) Rating Scales, and (1-5) Net Promoter Scales (NPS) (see Table 1).

5 FINDINGS AND RESULTS

We present our findings and results based on the experience reported by the participants who gathered the experimentation and evaluation of RetoñosApp— the methodological process and instruments used, were described in sections 3 and 4.

To analyze the User Experience (UX) of RetoñosApp, we contemplated some calculations (see Table 2), based on the average results by the participants of the questions posed in the questionnaire designed for the experimentation phase (see Table 1).

We sought to reach or exceed a threshold for User Experience of 65% acceptance by the participants of this pilot study. This indicates that end-users (i.e., target population) perceive an average user experience of 6.5 out of 10, which we could interpret as a perception of an acceptable User Experience.

We present on Table 3 the results obtained on the perception's evaluation, based on the questionnaire presented on section 3 (see Table 1). This analysis helped us identify how effectively the participants perceived RetoñosApp based on their experience through the experiment. From these results (see Table 3), the participants perceived a User-Experience of 8.16 out of 10, considered outstanding, 1.65 points above the expected result (i.e., 6.51 was expected). The participants evaluated the system usability (i.e., by calculating the SUS) with a score of 8.15, which is 2.16 points higher than expected (i.e., 5.99 was expected). Also, 81.81% of the participants would probably recommend RetoñosApp (i.e., by calculating the NPS), 31.81% above what was expected.

Moreover, from the comments and perceptions of the participants, the customized reports (i.e., educational roadmap) that RetoñosApp provides, encourage academic continuity and promote a deepened participation of students in their education (Baque-Reyes and Portilla-Faican, 2021). Also, we found that RetoñosApp promotes new learning and teaching strategies through the on-time customized "learning routes" provided to students and instructors— strategies aligned with defined curricular and methodological structures.

6 DISCUSSION

From the preliminary results and comments received by the participants of this pilot study, RetoñosApp effectively provides a friendly and user-centered asynchronous assistance and enhancement to learning processes, and feedback on teaching processes. Moreover, we also found that the preliminary participants' comments about their web-based platform experience were positive and constructive, guiding us to improve the tool on their comments for future iterations.

Unlike traditional algorithms, artificial intelligence (AI) does not seek to be 100% assertive regard-

Table 1: Questionnaire for the Perception's evaluation.

Question	Variable	Scale: SUS$^{(1)}$, NPS$^{(2)}$, RS$^{(3)}$
Q1: "I think I would like to use RetoñosApp more often in the course."	Usability	(1-5) SUS
Q2: "I found RetoñosApp to be unnecessarily complex to be used without prior instruction."	Usability	(1-5) SUS
Q3: "I thought RetoñosApp was easier to use than it really was."	Usability	(1-5) SUS
Q4: "I think I would need someone's technical support to be able to use RetoñosApp."	Usability	(1-5) SUS
Q5: "I found the features of RetoñosApp to be well integrated/developed according to what I would expect them to do."	Usability	(1-5) SUS
Q6: "I thought that there were too many inconsistencies, and contradictions in RetoñosApp in terms of the functionalities that I would expect the system to have in relation to its purpose."	Usability	(1-5) SUS
Q7: "I imagine that most people would learn to use RetoñosApp very quickly."	Usability	(1-5) SUS
Q8: "I found RetoñosApp very complicated to use."	Usability	(1-5) SUS
Q9: "I felt very confident or comfortable when using RetoñosApp."	Usability	(1-5) SUS
Q10: "I needed to learn a lot of things before I could use the RetoñosApp."	Usability	(1-5) SUS
Q11: "How precise, coherent and pertinent do you consider the functionalities provided by the application to be compared to the functionalities you expected?"	Usability	(1-10) RS
Q12: "How consistently do you think the graphic style (colors, shapes, images, and elements of the graphical user interface) of the application is concerning the institutional image of Universidad El Bosque?"	Usability	(1-10) RS
Q13: "How nicely do you think the application is in terms of its graphical user interface?"	Usability	(1-10) RS
Q14: "How complicated do you think it was for you to use the application?"	Usability	(1-10) RS
Q15: "How friendly do you think the application can be for any user?"	Usability	(1-10) RS
Q16: "How efficient do you think the application was regarding the number of errors it presented?"	Usability / Error rate	(1-10) RS
Q17: "How easy to learn to use do you think the application is?"	Usability	(1-10) RS
Q18: "How efficient do you think the application was during the ended user test?"	Usability	(1-10) RS
Q19: "How satisfied do you feel after using the application?"	Satisfaction	(1-10) RS
Q20: "How satisfied do you think you would be in the long term if the application were implemented as one more resource within the teaching and learning exercises of the subject?"	Satisfaction	(1-10) RS
Q21: "How much would you recommend your colleagues or friends use the application if there were modules for their respective interests, careers, subjects, or topics?"		(1-5) NPS
Q22: "How much would you recommend your instructor use the application as a complementary tool for the subject's methodology, fostering autonomous or independent study?"		(1-5) NPS

Notes: $^{(1)}$SUS: System Usability Scale, $^{(2)}$NPS: Net Promoter Score, $^{(3)}$ RS: Rating Scale

Table 2: Calculations of the contemplated metrics.

Metric	Calculation used
Usability (***U***)	$$U = Avg(Avg.Rating + Avg.SUS + Avg.NPS) \quad (1)$$ Avg.Rating corresponds to the average of questions 11-18, Avg.SUS corresponds to the average of questions 1-10, and Avg.NPS corresponds to the average of questions 21 and 22 (see Table 1).
User Satisfaction (***S***)	$$S = Avg(Q19 + Q20) \quad (2)$$ Q19 and Q20 correspond to questions 19 and 20, respectively (see Table 1).
User Experience (***UX***)	$$UX = Avg(U + S) \quad (3)$$ U and S correspond to the result of the metrics Usability and User Satisfaction, respectively. This calculation for User Experience (UX) is a preliminary approximation based on U and S calculated information, for this paper. Nevertheless, we are aware that UX contemplates usability (e.g., efficiency, perspicuity, dependability) and experience (e.g., originality, stimulation, interactivity) (Knijnenburg et al., 2012).

Table 3: Results of the perception's evaluation.

Metric	Expected Value	Score	Difference of the score vs. the minimum expected value
Usability [SUS]	8,15	2.14	5.99
Satisfaction [Rating Scales]	8,16	2.16	6.00
Recommendation [NPS]	81,81%	50%	31.81%
User-Experience (Usability & Satisfaction)	6.51	8.16	1.65

ing the results of its processes (Chen et al., 2020). Probabilistic factors determine the accuracy of AI models in CS education, since it learns from the context, environment, and users— all these elements are significantly malleable and volatile. Also, there is always a degree of uncertainty that affects the results when addressing teaching and learning strategies involving AI.

We evaluated the AI models part of RetoñosApp (i.e., Conversational Bot (NLP model) and Content-Based Recommendation System). As a result, the conversational bot had an accuracy of 82%, and the recommendation model had an accuracy of 83%. For these evaluation processes we used a preliminary N=40 training set. In the first place, both accuracies obtained by each model correspond accordingly as the content-based recommender model is naturally fed by the information collected by the platform (e.g., questionnaires and selections), and the conversational bot (e.g., questions, enriched answers, interaction). Moreover, we consider both integrated models' accuracy reliable as it conforms to the valid range of accuracies between 63.1% and 89.3% as reported in the Teaching Academic Survival Skill (TASS) community (Díaz-Galiano et al., 2019).

We find that a web-based tool, such as RetoñosApp, is a considerable way for users (i.e., CS students or instructors) who require accessibility, analysis, complement, and support to ease their teaching and learning processes in a customizable way. This is similar to how the LEGO (McNamara et al., 1999) model works, based on the participants' particular needs, doubts, and difficulties. Furthermore, these results also guided us to reflect on the difficulties perceived (e.g., how to improve the GUI in each module to get the most of the web-based tool), helping us identify elements to enhance for future iterations of RetoñosApp. In further research, a deepened analysis of the metrics that were calculated (see Table 2), will be addressed to complement the preliminary claims.

Therefore, we can preliminarily claim that RetoñosApp positively fosters the educational (i.e., teaching and learning) processes. The platform benefits both the teaching and learning processes of programming in CS. Moreover, unlike other available educational tools, RetoñosApp has the advantage that it can nurture the teaching and learning processes since it incorporates a conversational bot (i.e., virtual tutor) and a recommendation system (i.e., providing a customizable "educational roadmap" and frequent feedback). Further research on the tool will be conducted to support and complement all claims in this paper.

ACKNOWLEDGMENTS

The authors thank all the participants who voluntarily and actively collaborated in evaluating RetoñosApp. Also, the authors thank the Undergraduate Program of Systems Engineering instructors at Universidad El Bosque, Colombia, who willingly permitted this preliminary experimentation in their introductory programming courses in the early academic semesters.

REFERENCES

Adamopoulou, E. and Moussiades, L. (2020). An overview of chatbot technology. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 373–383. Springer.

Baque-Reyes, G. R. and Portilla-Faican, G. I. (2021). El aprendizaje significativo como estrategia didáctica para la enseñanza–aprendizaje.

Cano, P. A. O. and Alarcón, E. C. P. (2021). Recommendation systems in education: A review of recommendation mechanisms in e-learning environments. *Revista Ingenierías Universidad de Medellín*, 20(38):147–158.

Chen, L., Chen, P., and Lin, Z. (2020). Artificial intelligence in education: A review. *Ieee Access*, 8:75264–75278.

Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649.

Deacon, J. (2009). Model-view-controller (mvc) architecture. *Online][Citado em: 10 de março de 2006.] http://www. jdl. co. uk/briefings/MVC. pdf*, 28.

Díaz-Galiano, M. C., García-Cumbreras, M. Á., García-Vega, M., Gutiérrez, Y., Cámara, E. M., Piad-Morffis, A., and Villena-Román, J. (2019). Tass 2018: The strength of deep learning in language understanding tasks. *Procesamiento del Lenguaje Natural*, 62:77–84.

Grier, R. A., Bangor, A., Kortum, P., and Peres, S. C. (2013). The system usability scale: Beyond standard usability testing. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 57, pages 187–191. SAGE Publications Sage CA: Los Angeles, CA.

Harpe, S. E. (2015). How to analyze likert and other rating scale data. *Currents in pharmacy teaching and learning*, 7(6):836–850.

Harris, S. C. and Kumar, V. (2018). Identifying student difficulty in a digital learning environment. In *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, pages 199–201. IEEE.

Hassenzahl, M. and Tractinsky, N. (2006). User experience-a research agenda. *Behaviour & information technology*, 25(2):91–97.

Klein, M. H., Kazman, R., Bass, L., Carriere, J., Barbacci, M., and Lipson, H. (1999). Attribute-based architecture styles. In *Working Conference on Software Architecture*, pages 225–243. Springer.

Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., and Newell, C. (2012). Explaining the user experience of recommender systems. *User modeling and user-adapted interaction*, 22(4):441–504.

Mandal, P. C. (2014). Net promoter score: a conceptual analysis. *International Journal of Management Concepts and Philosophy*, 8(4):209–219.

Martin, R. C., Grenning, J., Brown, S., Henney, K., and Gorman, J. (2018). *Clean architecture: a craftsman's guide to software structure and design*. Number s 31. Prentice Hall.

McNamara, S., Cyr, M., Rogers, C., and Bratzel, B. (1999). Lego brick sculptures and robotics in education. In *1999 Annual Conference*, pages 4–369.

Moreno, J. J., Bolaños, L. P., and Navia, M. A. (2010). Exploración de modelos y estándares de calidad para el producto software. *Revista UIS Ingenierías*, 9(1):39–53.

Mutiawani, V. et al. (2014). Developing e-learning application specifically designed for learning introductory programming. In *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 126–129. IEEE.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Richards, M. (2015). *Software architecture patterns*, volume 4. O'Reilly Media, Incorporated 1005 Gravenstein Highway North, Sebastopol, CA . . . .

Saldarriaga-Zambrano, P. J., Bravo-Cedeño, G. d. R., and Loor-Rivadeneira, M. R. (2016). La teoría constructivista de jean piaget y su significación para la pedagogía contemporánea. *Dominio de las Ciencias*, 2(3 Especial):127–137.

Sommerville, I. and Torres, J. A. D. (2011). *Ingeniería de software*. Addison-Wesley Pearson Educación, México, 9 edition.

Wellnhammer, N., Dolata, M., Steigler, S., and Schwabe, G. (2020). Studying with the help of digital tutors: design aspects of conversational agents that influence the learning process.