American Journal of Educational Research, 2019, Vol. 7, No. 12, 976-982
Available online at http://pubs.sciepub.com/education/7/12/12
Published by Science and Education Publishing
DOI:10.12691/education-7-12-12

Towards a Web Application to Create Flowcharts for Supporting the Teaching-Learning Process of Structured Programming Courses

Fernando Vazquez-Peñaloza, Carlos R. Jaimez-González*

Departamento de Tecnologías de la Información, Universidad Autónoma Metropolitana, Ciudad de México, México
*Corresponding author: cjaimez@correo.cua.uam.mx

Received November 03, 2019; Revised December 06, 2019; Accepted December 26, 2019

Abstract Programming is an essential skill for students of undergraduate programs related to computer science or information technologies, because they are required to have the ability to generate algorithms that solve real problems, which are later implemented in a programming language. Flowcharts are diagrams that can be used to visualize algorithms, which are graphic representations of the processes that are involved in the algorithm; each process is represented with symbols, allowing the student a visual understanding of the solution for a given problem. This paper presents a web application to support the teaching-learning process of structured programming undergraduate courses, which allows the creation of flowcharts. The web application allows the teacher to create examples with their corresponding solutions, register exercises to be solved by students, which are provided with a work area to draw flowcharts by dragging and dropping elements.

Keywords: *structured programming, flowcharts, teaching-learning, web application*

Cite This Article: Fernando Vazquez-Peñaloza, and Carlos R. Jaimez-González, “Towards a Web Application to Create Flowcharts for Supporting the Teaching-Learning Process of Structured Programming Courses.” *American Journal of Educational Research*, vol. 7, no. 12 (2019): 976-982. doi: 10.12691/education-7-12-12.

1. Introduction

Programming is a necessary skill for the development of new information technologies that are becoming more common in daily life. Learning programming is not an easy task, since it requires an exhaustive study on specific topics related to it. Additionally, there are techniques to learn how to program, where different learning models are applied. Programming is a fundamental skill for students of undergraduate programs related to information technologies; this is because they are required to have the ability to generate algorithms that solve real problems, which are later implemented in a programming language. For this reason, undergraduate programs incorporate programming modules in their first year.

Students tend to have problems passing these subjects, failing or abandoning them prematurely, and this is due to the fact that they do not cover the necessary competencies to develop programs that demonstrate that the knowledge of the subject was acquired [1]. During the learning process of a programming language, students must face different challenges to demonstrate that they have the necessary skills to program. One of the main elements to take into consideration in order for a student to give the maximum performance and learn correctly depends on the teacher; he should be responsible for guiding and motivating students, in addition to applying different tools that complement the theoretical learning taught in the classroom [1].

The fact that a human being thinks differently than a computer, makes programming a difficult task for students that start to program [2]. First, students must understand how a human being thinks and then how a computer does it, this is where a technique is applied to transcribe real problems to a form in which one might think that a computer can understand it, which is achieved with an algorithm. At this stage is where the student must put into practice his knowledge on abstraction and logic for finding an appropriate solution for real problems.

Algorithms can be visualized in different ways, such as pseudo code and flowcharts; the latter are graphic representations of the processes that can be applied in the algorithm and each process is represented with symbols, allowing the student a visual understanding of the solution for the problems raised.

This project focuses on the development of a web application for the creation of flowcharts, such that these flowcharts serve as a tool to facilitate the learning of structured programming. The aim of the project is to provide students with more support tools for their learning and develop the ability to program in a dynamic way. On the other hand, the web application also helps teachers to present examples about the programming related to specific contexts, allowing them to clearly exemplify situations in which students require to use their problem-solving skills and at the same time understand the correct way in which an algorithm is solved by a computer.

The rest of this paper is organized as follows. Section 2 presents the theoretical framework, where the types of learning are covered, as well as the main elements that compose a flowchart are described. Section 3 covers the state of the art, where the main existing related tools are analyzed with a description of their features, a comparison table of the tools reviewed is also provided. Section 4 introduces the web application to create flowcharts, where its functionality is explained, some preliminary interfaces are shown, and the methodology with the activities carried out are listed. Finally, conclusions and future work are provided in section 5.

2. Theoretical Framework

One of the most notable differences that can be observed when determining if a student learns to program easier than another student, is the learning style; each individual is different and therefore requires different methods to properly get the knowledge necessary to acquire the programming skill. For this reason, it is necessary to know the different ways in which some teachers carry out the teaching process of programming at a higher level, and highlight the deficiencies that the different styles of teaching have at the time of applying them on the students.

The importance of the existence of a positive synergy between the student and the teacher of the subject is necessary for the correct teaching methods to be applied, as discussed in [3] and [4], in which the types of learning can be divided in four categories:

1. *Active*, they learn by manipulating things and working with others; or *Reflective*, they learn by thinking about things and working alone.

2. *Sensitive*, they are concrete, practical, oriented towards facts and procedures; or *Intuitive*, they are conceptual, innovative, oriented towards theories.

3. *Visual*, they prefer the visual presentation of materials such as films, tables, or flowcharts; or *Verbal*, they prefer written or spoken explanations.

4. *Sequential*, they learn little by little in an orderly way; or *Global*, they learn in a general way integrating their previously acquired knowledge.

The results of the research carried out in [4] showed that the majority of people are *Active*, *Sensitive*, *Visual* and *Sequential*; they prefer to interact with others to obtain knowledge, in addition to applying procedures in an orderly way, using examples that students can visually perceive, such as diagrams, videos and images.

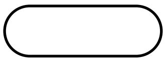
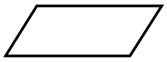
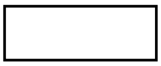
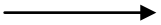
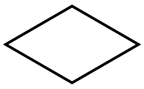
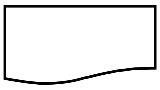
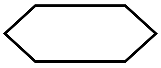
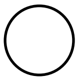
Taking into consideration the results presented by the authors of [4], this project focuses on the design of a solution as a web application, using a software engineering methodology [5], where there are different steps for the creation of a computer program: analysis, design, implementation and testing.

In the analysis process, the necessary information that the developer must know about the main requirements of the system to be developed is collected, as well as special features that are to be taken into account so that the information system covers the needs raised. Afterwards, the methodology continues with the design process, this is where the developer has multiple options to design an algorithm that can solve the problem. This process is based on the specifications previously defined in the analysis stage, and the design can be developed in a flowchart or in pseudo code [2].

Flowcharts are visual representations of algorithms, where computational processes are represented by symbols and are related through arrows, which represent the flow of the algorithm. The symbols used have been regulated by the American National Standards Institute (ANSI) [2, 6]. Table 1 represents the main processes used within flowcharts.

These elements used in flowcharts help to visualize the processes, which facilitates the understanding of the flow of data that a program receives, and how through the flow it can take different paths and processes depending on the design of the solution. After designing the solution using a flowchart, you can choose to use pseudo code, which is a combination of natural language, logical statements and arithmetic operations to design a more specific solution and its implementation in a programming language.

Table 1. Main symbols used in flowcharts

	It is used to indicate the start and the end of the flowchart.
	It is used to represent a data input (reading).
	It is used to represent processes, such as arithmetic operations, value assignments in memory, etc.
	It is used to indicate the flow direction of the diagram.
	It is used to indicate a decision process. The condition to be compared is described within the form.
	It is used to indicate the output (printing) of data.
	It is used to indicate a cycle of repetitions of an action.
	It is used to indicate the continuation of flow of the diagram within a same page.

These steps are normally needed to create a computer program and are basic to learn correctly the technique for programming. Additionally, as mentioned in [7], flowcharts provide novice programmers with the logical understanding of an algorithm; for this reason, they are used in the teaching of programming, as they are useful as reinforcement to the knowledge acquired in classes.

On the other hand, the motivation that the teacher provides is vital in order for the students to give their maximum performance when learning the subject. Additionally, the methods and tools that the teacher uses in classes should be related to the previously mentioned learning styles.

As mentioned in the research carried out in [1], teachers apply different methods for the teaching of programming subjects such as laboratory practices, exercises inside and outside classes, workshops, problem solving, research on the Internet, audiovisual exhibitions, among others.

A survey carried out with teachers [1] highlights that among the different methods and tools used to teach the subject, the exercises outside the classroom do not meet the minimum requirements for learning the subject, this is due to the lack of motivation of the students for self-learning. Based on this premise, the present project will focus on presenting a solution to this problem related to work outside the classroom, reinforcing the knowledge acquired in classes through a web application.

A web application can integrate different services that offer the Internet through a hypertext language; this application can communicate with multiple users to exchange specific information [8]. This information is presented to the users of the web application through an interface that is deployed to web browsers, which act as the means by which end users can interact with the information contained in the different applications. The web application must be created with an interpreted programming language, which means that the source code is analyzed and executed directly on a web browser, which is the interface that the user uses to deploy the web application [9].

A web application is composed of an ordered and related set of elements that, under a specific operation, return information processed through an application database or through the collection and processing of information from external data sources [8]. Users will interact with the application with the intention of retrieving information; however, there is the possibility that the user becomes the author of the content that appears in the application, this is achieved through a dynamic web application design, in which the user can modify the content using the elements that are provided in the interface.

3. State of the Art

This section analyses different existing systems to create flowcharts, their features, as well as some screenshots that show their functionality. A comparative table will also be presented, which takes into consideration the most important features of the existing systems reviewed.

3.1. PSeInt

PSeInt [10] is a desktop system that helps students in the coding of computer programs and algorithms, mainly for writing pseudo code in an environment with different teaching tools, such as writing, auto correction, pop-up windows for help, coloring of structures, coloring of syntax, among others.

It has a tool that allows the student to create a flowchart with the main forms; it provides an important feature, which is the facility to represent repetitive structures, conditional structures and create variables to perform logical and arithmetic operations. Figure 1 shows on the left side of the screenshot a flowchart with a basic example, where two variables are entered by the user and the result of the sum is displayed. The right side of the screenshot shows the different elements that can be incorporated to the flowchart.

Once the flowchart has been created, the system will automatically create in natural language the pseudo code of the designed program represented in the flowchart. In order to visualize the algorithm created through the flowchart, it has a tool that executes the pseudo code generated in a command prompt window, demonstrating the behavior of the previously described operations. Figure 2 shows the screenshot of the system with the corresponding pseudo code for the flowchart created in Figure 1. On the left side window, the pseudo code is provided, while in the right side panel there are some buttons to start and pause the execution of the program generated.

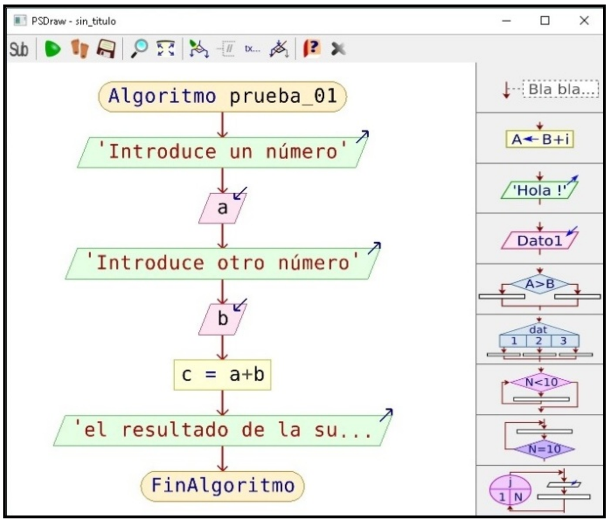
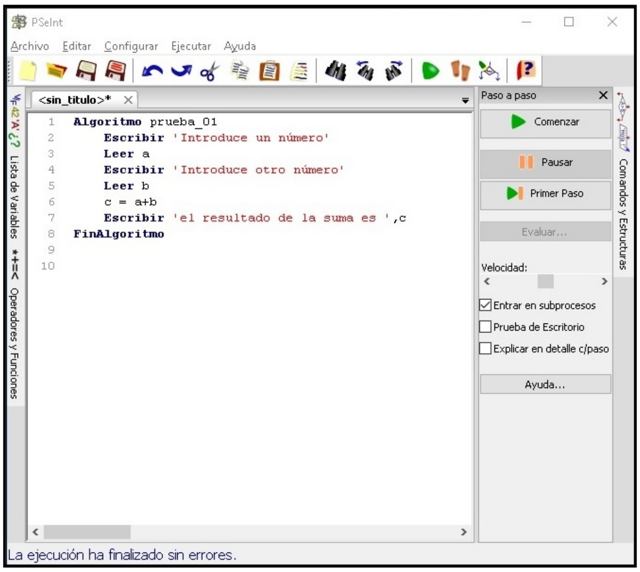


Figure 1. Flowchart created with PSeInt



```
Algoritmo prueba_01
	Escribir 'Introduce un número'
	Leer a
	Escribir 'Introduce otro número'
	Leer b
	c = a+b
	Escribir 'el resultado de la suma es ',c
FinAlgoritmo
```

☑ Entrar en subprocesos
☐ Prueba de Escritorio
☐ Explicar en detalle c/paso

La ejecución ha finalizado sin errores.

Figure 2. Pseudo code generated with PSeInt

3.2. Dia

Dia [11] is a program to design structured diagrams, it is inspired by the Windows program Visio, but oriented to the creation of diagrams for casual use. It is designed to create flowcharts; however, it is also possible to create other types of diagrams, such as UML diagrams [12], entity relationship diagrams, network diagrams, among others.

In order to create flowcharts, it has the main figures used in flowcharts, in addition to other forms to represent more specific processes. Figure 3 shows the different processes included in *Dia*, for the creation of flowcharts.

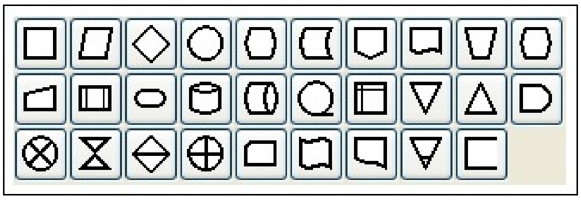


Figure 3. Processes for the creation of flowcharts in Dia

It allows users to export the flowcharts created to different output formats, such as pdf, jpg, png, among others. Figure 4 shows a flowchart created with *Dia*; the different processes or forms are shown in the left panel, and the flowchart that is being created is shown in the right panel.

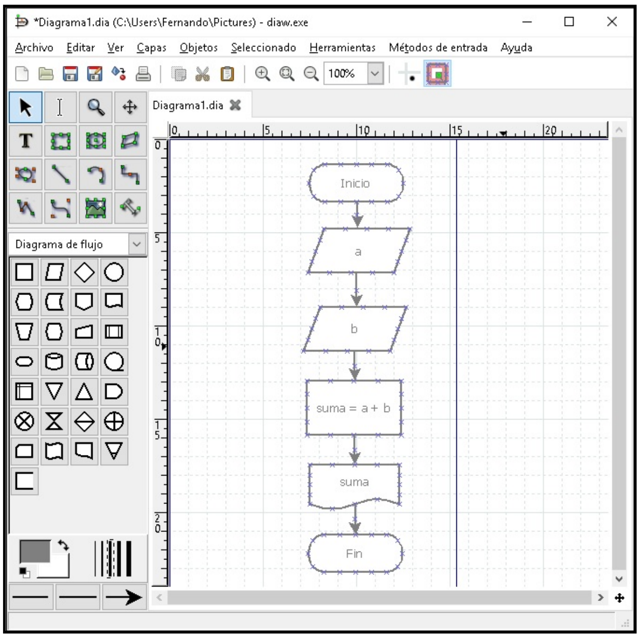


*Diagrama1.dia (C:\Users\Fernando\Pictures) - diaw.exe
Archivo Editar Ver Capas Objetos Seleccionado Herramientas Métodos de entrada Ayuda
100%
Diagrama1.dia
Diagrama de flujo
Inicio
a
b
suma = a + b
suma
Fin

Figure 4. Flowchart created with Dia

3.3. Draw.io

Draw.io [13] is an open web platform for the creation of different types of diagrams. It contains the most common forms for the creation of flowcharts.

It has a tool to divide the diagram into layers, allowing the diagram to be divided into different sections and allowing to modify the size and color of the background. Figure 5 shows the interface of *Draw.io* and an example of the creation of flowcharts; the left panel contains the different forms that can be used to draw the flowchart, and the right panel has an area to draw the flowchart with the elements from the left panel.

It allows you to share the flowcharts through a URL or directly to social networks, such as *Facebook* or *Twitter*. Once shared, the flowchart can be modified by the new user. It also allows to save flowcharts in the cloud or export them to specific formats such as png, jpeg, pdf, among others.

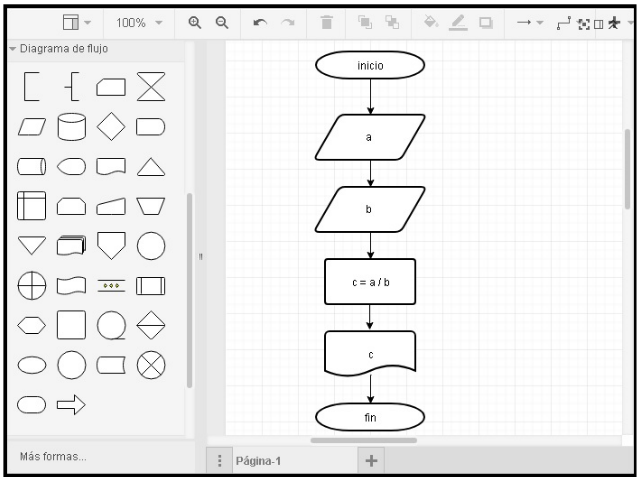


Figure 5. Flowchart created with Draw.io

3.4. LucidChart

LucidChart [14] is a web system that allows the creation of flowcharts, but also allows the creation of other types of diagrams such as relationship, flow, data, organization, network infrastructure, among others. For the elaboration of flowcharts, it contains the basic forms, and has a tool to change the style of the text of the diagrams, as well as the color of the figures.

Figure 6 shows the interface of *LucidChart* and an example of the creation of a flowchart; on the left panel it has the different forms to be used in flowcharts; in the right panel it shows the flowchart that has been created with the elements taken from the left panel.

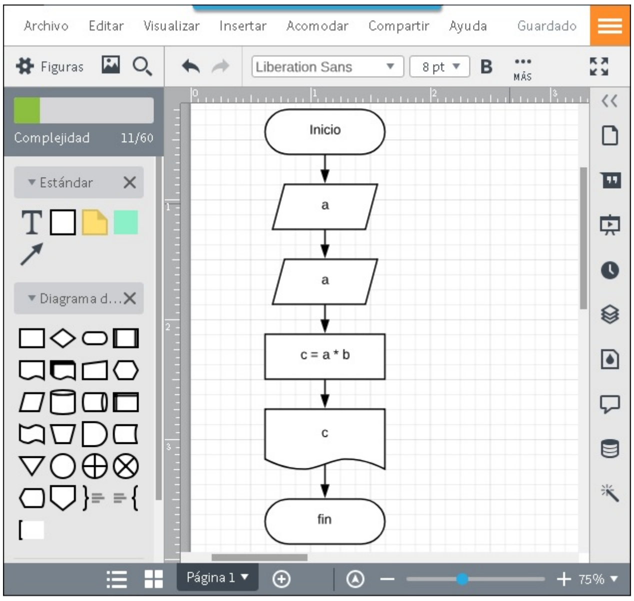
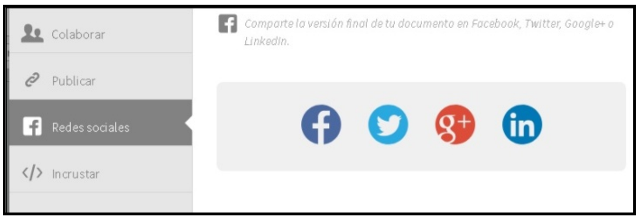


Figure 6. Flowchart created with LucidChart

It allows to share the flowcharts through a specific URL that will contain the flowchart, and users who enter the link will be able to visualize the flowchart. In addition, they may request permission from the author of the diagram to modify it. It also allows to share the flowchart in the main social networks, such as *Facebook*, *Twitter*, *Google+*, among others. Figure 7 shows the different options that the users have to share their flowcharts.



Comparte la versión final de tu documento en Facebook, Twitter, Google+ o LinkedIn.

Figure 7. Options to share a flowchart in LucidChart

3.5. Cacoo

Cacoo [15] is a website dedicated to the creation of different diagrams and mockups. It has the basic forms for the creation of flowcharts and tools to provide styles to the figures in the diagrams. It allows to export the diagrams to different output formats, such as pdf, jpeg, png, among others.

It allows to share the flowcharts through social network accounts or directly from a URL generated for the created diagram. Figure 8 shows the interface for creating a flowchart in *Cacoo*, where the left panel has the different options for incorporating forms to the flowchart, importing, exporting, printing, sharing, among others; the right panel shows the flowchart created.

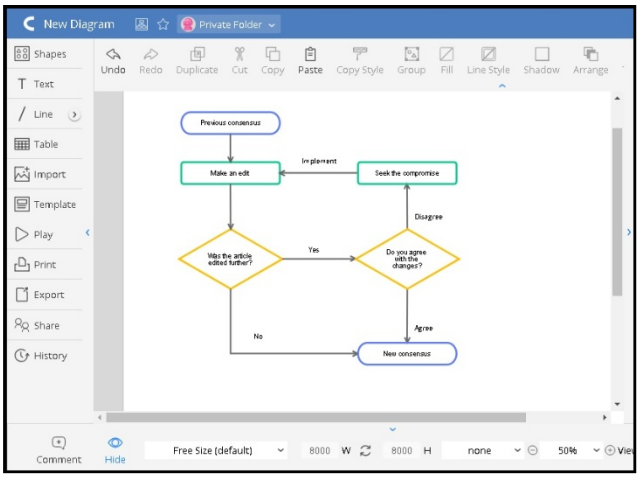


Figure 8. Flowchart created with Cacoo

3.6. Comparative Table

Table 2 shows a comparison of the features offered by the systems reviewed in the previous sections: T1) *PSeInt*, T2) *Dia*, T3) *LucidChart*, T4) *Draw.io*, T5) *Cacoo*. A tick indicates that the system has the feature, while a cross indicates that the system does not have it. A brief description of the features is also provided.

Table 2. Comparison of features in the existing systems

Features	T1	T2	T3	T4	T5
Online system	×	×	✓	✓	✓
Drag and drop	✓	✓	✓	✓	✓
Save files	✓	✓	×	×	×
Export to image	✓	✓	✓	✓	✓
Pseudo code	✓	×	×	×	×
Store in cloud	×	×	✓	✓	✓
Share diagram	×	×	✓	✓	✓

Online system. This feature means that the system is available online.

Drag and drop. This feature means that the system allows to drag and drop forms to a work area to create a flowchart in the system.

Save files. This feature means that the system allows to store the flowcharts created as files, in order to be retrieved and modified later.

Export to image. This feature means that the system allows to export the flowcharts created as images, such as jpg, png, among others.

Pseudo code. This feature means that the system automatically generates a pseudo code representation from the flowchart created.

Store in cloud. This feature means that the files stored in the system are available in the cloud.

Share diagram. This feature means that the system allows to share the diagrams created in the main social networks, such as *Facebook*, *Twitter*, among others.

4. The Web Application

This section describes the web application to create flowcharts, which is currently under development. The design of the web application is divided into modules for easy understanding and development. The functionality of the web application is described in this section, the main objectives, as well as the methodology to be carried out in order to complete its development.

4.1. Functionality

The web application includes an authoring web system, where one of the users, the teacher, can design practical exercises and have the possibility of saving them, in order to share them with other users, the students. One of the main features of the web application to create flowcharts is that the teacher can create examples for the students. Some of the functionalities of the web application are listed in the following paragraphs:

- A section to create a practical problem. The aim is that the teacher can describe the practical problem with natural language; this problem must be solved in a structured way and with the main elements provided by a standard flowchart. The problem can be saved to the web application, with the aim of generating examples of more specific topics to reinforce the learning of the students.

- A section for the student to solve the problems and exercises proposed by the teacher. This section is where students are able to drag and drop the different forms to create a flowchart to solve a specific problem given by the teacher previously.

- A section for the student to freely draw flowcharts. The aim of this section is to provide the student with an area to practice how to create a flowchart, which is not necessarily related to a problem or exercise given by the teacher. This area can be used at any time to draw a flowchart, which can also be saved.

- A section to review practical examples related to structured programming. This section is created by the teacher with the aim of providing students a set of examples to study structured programming. The examples contain a description of the problem and the solution with an associated flowchart.

A preliminary version of the web application interface is shown in Figure 9, where the student can create flowcharts freely. The top panel has three options: *create a diagram*, which allows the student to draw a flowchart; *exercises*, which includes exercises created by the teacher for the student to solve; and *examples*, which allows the student to review practical examples related to structured programming. The left panel includes the forms that can be dragged and dropped on the work area that is shown in the right panel, where a flowchart is being created; the left panel also includes some style options to change the font family, font size and font color.

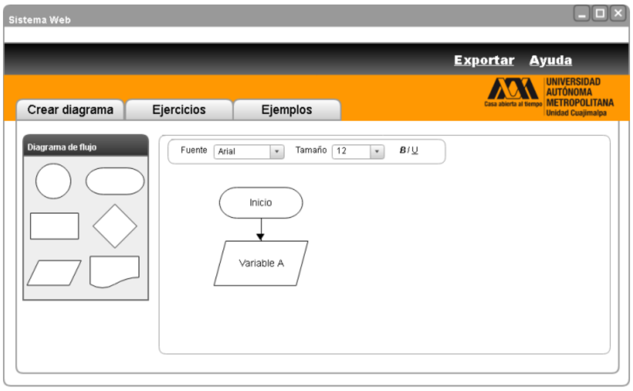
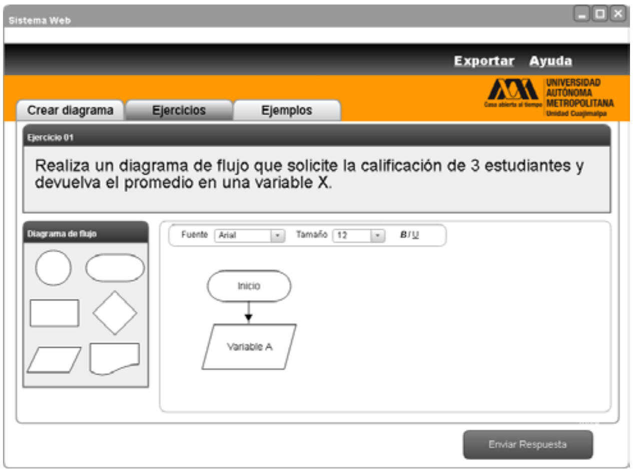


Figure 9. Interface for drawing flowcharts freely



Realiza un diagrama de flujo que solicite la calificación de 3 estudiantes y devuelva el promedio en una variable X.

Figure 10. Interface for solving exercises

Figure 10 illustrates the interface for *exercises*, where the students can solve the problems proposed by the teacher. The top panel provides a description of the problem to be solved by the students; the left panel contains the forms to be dragged and dropped to the work area; the right panel contains the work area, where the student draws the flowchart to give an answer to the problem; finally, at the bottom of the screen there is a button to send the flowchart to the teacher, when the student has finished drawing the flowchart.

It should be noticed that the web application provides the teacher a section to mark the problems solved by the students, where the teacher has the possibility of providing written feedback about the flowchart drawn by the student, and how well this represents a solution for the problem given.

The main objective of this project is to develop a web application to create flowcharts that reinforce the learning of structured programming in students who start an undergraduate program related to computer science or information technologies. The particular objectives of the project are the following:

- Develop an interface for the student to understand the meaning of each symbol used to draw flowcharts.

- Develop a mechanism for the teacher to create practical examples within the web application.

- Develop a mechanism that allows students to solve problems through the creation of flowcharts within the web application.

- Develop a mechanism to export the flowcharts to a specific image format, such as jpg, png, among others.

- Develop a feedback mechanism, where the teacher can provide comments to the student in relation to the flowcharts created.

4.2. Methodology

The web application is divided in modules, which represent the activities that are being carried out throughout the development of the project. The following paragraphs summarize some of the activities that compose the methodology of this project.

Development of an interface for practical exercises. This interface takes into account the needs of a teacher to propose practical exercises and the tools that the student requires in order to generate a flowchart for the solution.

Development of an interface for examples of structured programming. This interface allows the student to solve basic exercises of structured programming.

Design of example exercises for the introduction of structured programming. Example exercises are designed based on some books that contain practical activities related to structured programming.

Storage of example exercises for the introduction to structured programming. The exercises are stored in the web application, which are part of the interface for examples of the introduction of structured programming.

Development of the database. A database has been created with the objective of storing the work that the teacher carries out with the students, the possibility that each diagram created will be a SVG file will be considered.

Development of SVG images in the work area of the web application. SVG images will be developed from images previously stored in the web application.

Development of functionality for interaction between images. Functions are implemented for the SVG elements created for its correct functionality within the web application.

Development of functionality for image manipulation. The functionality to drag, drop and resize the images in the work area will be implemented.

Development of storage mechanism. A mechanism is developed to store and retrieve the flowcharts created, considering the SVG format.

Development of a mechanism to publish diagrams. A mechanism will be developed to publish the diagrams from a URL.

Development of an interface for feedback to the student of practical exercises. An interface will be developed so that the teacher can review and evaluate the flowcharts stored by the students.

Evaluation of the web application. An evaluation of the functionality and perception of the application will be carried out with the teachers and students.

5. Conclusions and Future Work

This paper presented a web application to support the teaching-learning process of structured programming undergraduate courses, which allows the creation of flowcharts. The web application allows the teacher to create examples with their corresponding solutions, register exercises to be solved by students, which are provided with a work area to draw flowcharts by dragging and dropping elements.

A comparison of five existing related tools was carried out, where different features were analyzed, such as the possibility of executing it online, dragging and dropping elements, saving files to the cloud, exporting the flowcharts as images, generating automatically pseudo code, sharing diagrams, among others. The results of the comparison showed that none of the tools were designed to support the teaching-learning process of structured programming courses.

Further work is needed to complete the functionality of the web application. Usability and functionality tests will be carried out with students of the structured programming course [16], which belongs to the undergraduate program in Information Technologies and Systems [17] of the Metropolitan Autonomous University, Cuajimalpa Campus, in Mexico. It is also planned to evaluate the web application through an evaluation instrument that will measure aspects of its interface design, its usefulness and its didactic features. The results of this evaluation will be considered for improving the web application.

References

[1] Beltrán, J., Sánchez, H., Rico, M. (2015). Análisis cuantitativo y cualitativo del aprendizaje de Programación I en la Universidad Central del Ecuador. Revista Tecnológica de la ESPOL, 28 (5), pp. 194-210.

[2] Corona Nakamura, M. A., Ancona Valdez, M. (2011). Diseño de algoritmos y su codificación en lenguaje C. México: McGraw-Hill.

[3] Figueroa, N., Cataldi, Z., Méndez, P., Rendon Zander, J., Costa, G., Salgueiro, F. A., Lage, F. J. (2005). Los estilos de aprendizaje y el desgranamiento universitario en carreras de Informática. Jornadas de Educación en Informática y TICs en Argentina.

[4] Muñoz, R., Barría Martínez, M., Nöel, R., Quiroz, P. (2012). Determinando las Dificultades en el Aprendizaje de la Primera Asignatura de Programación en Estudiantes de Ingeniería Civil Informática. Congreso Internacional de Informática Educativa.

[5] Jacobson, I., Booch, G., Rumbaugh, J. (1999). The Unified Software Development Process. Boston, Massachusetts, USA: Addison-Wesley Professional.

[6] Joyanes Aguilar, L. (2008). Fundamentos de Programación: Algoritmos, estructura de datos y objetos. Madrid, Spain: McGraw-Hill.

[7] Supaartagorn, C. (2017). Web application for automatic code generator using a structured flowchart. 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China.

[8] Rodríguez Perojo, K., Ronda, R. (2006). El web como sistema de información. Acimed, 14 (1), pp. 1-15.

[9] Arbeláez Salazar, O., Medina Aguirre, F. A., Chaves Osorio, J. A. (2011). Herramientas para el Desarrollo Rápido de Aplicaciones Web. Scientia Et Technica, 17 (47), pp. 254-258.

[10] Novara, P. (2018). PseInt. A tool for supporting students in their first steps to structured programming. Available: http://pseint.sourceforge.net/.

[11] Cherepii, S. (2018). Dia. Program for diagram creation. Available: https://wiki.gnome.org/Apps/Dia

[12] Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modelling Language User Guide. Boston, Massachusetts, USA: Addison-Wesley Professional.

[13] Draw.io (2018). Flowchart Maker and Online Diagram Software. Available: https://www.draw.io/

[14] LucidChart (2018). Collaborative diagramming solution. Available: https://www.lucidchart.com

[15] Nulab (2018). Cacoo. Creator of diagrams based on the cloud. Available: https://app.cacoo.com

[16] Metropolitan Autonomous University, Cuajimalpa Campus (2018). Syllabus of Structured Programming. Information Technologies and Systems Undergraduate Program. México: UAM. Available: http://www.cua.uam.mx/pdfs/lic/tsi/n2/460005_programacion_estruc.pdf.

[17] Metropolitan Autonomous University, Cuajimalpa Campus (2018). Information Technologies and Systems Undergraduate Program. México: UAM. Available: http://hermes.cua.uam.mx/archivos/PlandeEstudioTSI.pdf.