

# HERRAMIENTAS PARA LA IMPARTICIÓN DE CLASES DE PROGRAMACIÓN EN MODALIDAD A DISTANCIA

## TOOLS FOR THE TEACHING OF PROGRAMMING CLASSES IN REMOTE MODE

A. D. Nieto Yáñez<sup>1</sup>  
R. E. López Díaz<sup>2</sup>  
I. Y. Hernández Báez<sup>3</sup>  
S. E. León Sosa<sup>4</sup>

### RESUMEN

En este artículo se presenta el uso y análisis de diferentes herramientas usadas para impartir asignaturas como Programación, cuyas horas prácticas en el plan de estudios son el doble de las horas teóricas y que requerían el trabajo del estudiante en un laboratorio de cómputo tanto de forma individual como colaborativa. Las herramientas empleadas corresponden a cada uno de los pasos para resolver un problema mediante la implementación de un programa, hasta la fase de depuración y pruebas. La teoría se siguió explicando de forma parcial con una presentación de diapositivas y la parte práctica se trabajaba con una pizarra compartida, una herramienta para realizar y compartir algoritmos de flujo y un compilador que permita trabajar en línea y compartir archivos. Adicional a estas herramientas se usó Google Classroom y Google Meet. La principal ventaja de las herramientas usadas es el poder compartir y trabajar de forma colaborativa en el proceso de enseñanza y aprendizaje de Programación.

### ABSTRACT

This article presents the use and analysis of different tools used to teach subjects such as Programming, whose practical hours in the study plan are twice the theoretical hours and which required the student's work in a computer laboratory both individually and collaborative. The tools used correspond to each of the steps to solve a problem through the implementation of a program. The theory continued to be partially explained with a slide presentation and the practical part was worked with a shared whiteboard, a tool to create and share flow algorithms and a compiler that allows working online and sharing files. In addition to these tools, Google Classroom and Google Meet were used. The main advantage of the tools used is the ability to share and work collaboratively in the teaching and learning process of Programming.

### ANTECEDENTES

Dentro del plan de estudios de Ingeniería en Tecnologías de la Información de la Universidad Politécnica del Estado de Morelos, el cual puede considerarse una actualización del plan de Ingeniería en Informática se incluyen asignaturas relacionadas con la programación, desde Introducción a la Programación, Programación (Programación Estructurada usando lenguaje C) y Programación Orientada a Objetos, entre otras. Estas asignaturas representan uno de los ejes de formación de la carrera y, de acuerdo con el estudio de empleabilidad de la Universidad un gran porcentaje de los egresados tienen un empleo en esta área, esto genera

---

<sup>1</sup> Profesora Investigadora de la carrera de Ingeniería en Informática de la Universidad Politécnica del Estado de Morelos.  
[anieto@upemor.edu.mx](mailto:anieto@upemor.edu.mx)

<sup>2</sup> Profesor Investigador de la carrera de Ingeniería en Informática de la Universidad Politécnica del Estado de Morelos.  
[rlopezd@upemor.edu.mx](mailto:rlopezd@upemor.edu.mx)

<sup>3</sup> Profesora Investigadora de la carrera de Ingeniería en Informática de la Universidad Politécnica del Estado de Morelos.  
[ihernandez@upemor.edu.mx](mailto:ihernandez@upemor.edu.mx)

<sup>4</sup> Profesora Investigadora de la carrera de Ingeniería en Informática de la Universidad Politécnica del Estado de Morelos.  
[lsandra@upemor.edu.mx](mailto:lsandra@upemor.edu.mx)

la importancia de que la impartición de estas asignaturas sea de manera práctica y con la misma calidad que se tenía en la modalidad presencial, pero, ahora en la modalidad virtual.

En la modalidad presencial dentro de la universidad, estas asignaturas se impartían en un laboratorio de cómputo en el que cada estudiante tenía a su disposición una computadora (todas con capacidades similares) que tenía instalados diferentes entornos de desarrollo para las diferentes asignaturas. Adicionalmente, se usaba Classroom o drive, pero, sólo para entrega de evidencias y compartir material.

Aunque cada profesor o profesora de estas asignaturas puede usar técnicas de enseñanza diferentes, mediante acuerdos de academia se establecen que las asignaturas se deben evaluar usando un portafolio que incluye evidencias de producto, de desempeño y de conocimiento y un determinado porcentaje de estas evidencias debe ser obtenido de forma individual y otro en equipo. Este tipo de evaluación y la realización de evidencias formativas en el laboratorio en el cual el profesor podía observar y apoyar a los estudiantes se complicó al pasar de una modalidad 100% presencial a una completamente a distancia, por lo cual se requería usar herramientas que sustituyeran o complementaran a las diapositivas usadas para explicar la teoría, el compilador y el papel y lápiz.

Otra problemática es la desmotivación de los estudiantes que claramente prefieren la modalidad presencial y que los distractores aumentaron en la mayoría de los casos con el cambio a educación a distancia, según Biggs (2004) para la docencia resulta un desafío que los estudiantes se comprometan y motiven con las actividades de aprendizaje y usar las Tecnologías de Información y Comunicación (TIC) de acuerdo con Sánchez (2003), pueden ser una fuente de innovación para la enseñanza, contribuyendo al aprendizaje, ofreciendo metodologías y recursos para el estudiante del siglo XXI.

Adicional a esto, la universidad optó por continuar con la impartición de clases de manera síncrona, que de acuerdo con Mujica (2020) permite la interacción de los estudiantes con el profesor y con sus compañeros, también permite que se resuelvan de manera instantánea sus dudas y que los estudiantes no sólo vean, sino realicen las actividades.

El incorporar herramientas síncronas y asíncronas en un curso es una tarea muy importante y como se menciona en Santoveña (2012), esta debe ser respaldada por un estudio previo, sin embargo, las circunstancias actuales impidieron que un estudio pudiera realizarse y la incorporación de herramientas fue necesaria para continuar con la impartición de clases.

En este artículo se presenta el análisis de las herramientas usadas para impartir asignaturas de programación. Se hace énfasis en los pasos para resolver un problema mediante la implementación de programas de computadoras. Desafortunadamente, no se encontró una herramienta que pueda apoyar en todas las actividades, por lo cual surge la necesidad de seleccionar varias herramientas que se complementen.

El objetivo de este trabajo es analizar el uso de herramientas que puedan apoyar en el trabajo individual y en equipo de los estudiantes que cursan una asignatura de programación.

Como se mencionó anteriormente, la impartición de estas asignaturas debe continuar con evidencias formativas y sumativas realizadas de manera individual y en equipo y debido a que el cambio drástico de presencial a virtual impidió que se siguieran realizando de la manera habitual fue necesario analizar y proponer herramientas útiles para que los estudiantes alcanzaran los resultados de aprendizaje de cada asignatura.

## METODOLOGÍA

La revisión de herramientas que podían ser utilizadas se realizó de manera gradual, de acuerdo con los resultados de aprendizaje se veía la necesidad de incorporar nuevas herramientas para solventar los problemas detectados. Para cada problema detectado se seguían los siguientes pasos para elegir la herramienta:

1. Revisar herramientas que apoyaran a resolver la problemática encontrada. Analizar ventajas y desventajas, prefiriendo aquellas en que los estudiantes no requieran instalar o algún dispositivo adicional y que fuera fácil de usar para ellos. Si las herramientas no son gratuitas, revisar si se tiene una versión gratis y si esta versión es suficiente para la actividad a realizar.
2. Instalar o registrarse (en caso de que fuera en línea).
3. Usarla en una clase y recibir retroalimentación mediante una encuesta informal y para realizar la supervisión de la actividad diseñada, se recabó información a través de la observación directa:
  - a. Registrando el comportamiento de los estudiantes,
  - b. Su interés por las herramientas utilizadas,
  - c. Las dudas que presentaron durante el desarrollo de la actividad
  - d. Y los comentarios generales durante la sesiónAdemás, se recabaron los ejercicios resueltos por los alumnos
4. En caso de que la retroalimentación fuera negativa probar con otra alternativa.

Para identificar las herramientas que se requerían para reemplazar el pizarrón y el acompañamiento del profesor en el laboratorio se analizó la metodología usada para la resolución de problemas mediante la computadora, esta metodología consta de un proceso de siete pasos como se muestra en la Figura 1.

Para el desarrollo de este trabajo se planteó proponer herramientas para los primeros cinco pasos, esto incluye desde la definición del problema hasta la fase de depuración y pruebas. La documentación es parte importante y se realiza mediante comentarios en el código fuente y, posteriormente, un reporte del desarrollo, por lo que respecta al mantenimiento esta etapa no es abordada en los primeros cursos de programación por lo cual no fue considerada en este trabajo.

Las fases son las siguientes:

1. Definición del problema: Es el enunciado del problema, el cual debe ser claro, completo, con alcances y limitaciones. Este paso es elaborado por el profesor y explicado.
2. Análisis del problema: En este paso se analizan entradas y salidas, así como el proceso general para que dadas las entradas se obtenga la solución deseada.
3. Diseño de la solución: Una vez definido y analizado el problema, se produce a la creación del algoritmo, en este caso se considera el uso de diagramas de flujo.

4. Codificación: Consiste en escribir la solución del problema en una serie de instrucciones detalladas en un código reconocible por la computadora, en este caso es lenguaje C.
5. Prueba y depuración: Prueba es el proceso de identificar los errores que se presenten durante la ejecución de programa y se deben emplear entradas correctas, entradas incorrectas y analizar las salidas. La depuración consiste en eliminar los errores que se hayan detectado durante la prueba, para dar paso a una situación adecuada y sin errores.



**Figura 1.** Metodología para la resolución de problemas usando computadora

**RESULTADOS**

En el primer curso de programación iniciado en la pandemia se usaron las siguientes herramientas:

- Curso en Google Classroom que se decidió seguir utilizando porque era la forma en la que se compartía material con los estudiantes y ellos entregaban sus evidencias antes de la pandemia. En la Figura 2, se muestra un ejemplo de un curso. Además del uso previo, se usó para realizar encuestas y aplicar evidencias de conocimiento usando formularios de Google, se usaba el muro para dar avisos y se mantenía comunicación para resolver dudas mediante mensajes privados.



**Figura 2.** Curso de Programación en Classroom

En la Figura 3 se muestra la pantalla de trabajo de clase donde se puede ver que se aplicó una evidencia de conocimiento usando un formulario. Se aplicaron cuestionarios diferentes con la misma dificultad y las preguntas se mostraban en orden aleatorio.

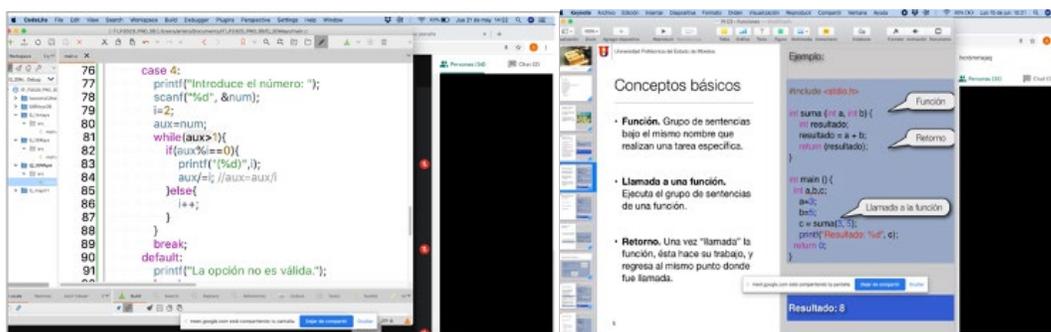


**Figura 3.** Curso de Programación en Classroom, aplicando un formulario de Google

- Reuniones en Google Meet para tener una clase de forma síncrona. En la Figura 4, se muestra una reunión en Meet, la cual inicialmente se tenía que generar de manera separada porque no estaba activada en Classroom, así que se compartía la liga a la reunión a través del muro.

En estas llamadas se compartía la pantalla y los estudiantes podían ver el proceso para resolver un problema, para la teoría se usaban diapositivas, al igual que para el análisis del problema. El diseño se realizaba usando diapositivas o archivos en Word o Excel, dependiendo del problema y se realizaba algoritmo o pseudocódigo. La codificación y depuración se realizaba usando Codelite, el cual era visto por los estudiantes a través de la pantalla compartida y fragmentos a través del chat y en algunas ocasiones a través de la publicación de material en el curso de Classroom (además de la grabación de la clase).

Para las actividades, los estudiantes tenían instalado el compilador de C y algunos trabajaban con un compilador en línea, debido a que tenían limitaciones en el dispositivo que usaban para atender las clases. Para apoyar a los estudiantes en la realización de actividades, los estudiantes compartían su pantalla, lo cual era un proceso tardado, ya que, implicaba descompartir, que ellos compartieran y descompartieran, para poder atender a otro estudiante.

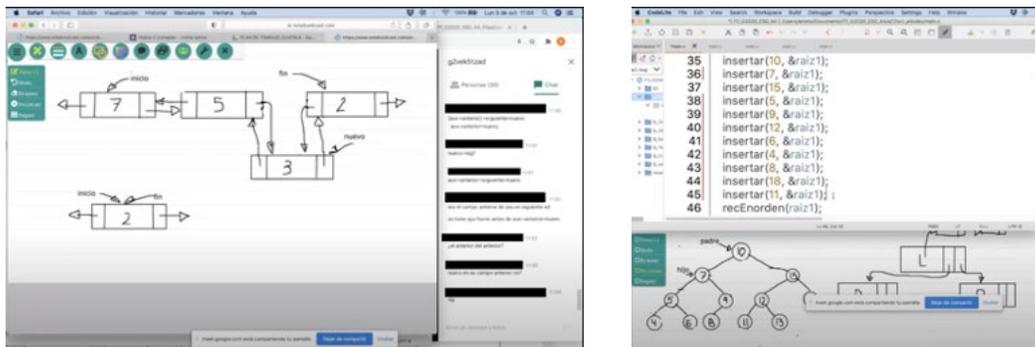


**Figura 4.** Uso de una reunión de Meet para clase síncrona. En la izquierda compartiendo la pantalla del compilador y a la derecha, unas diapositivas

La principal desventaja era que muchos de los estudiantes requerían que las actividades fueran realizadas paso a paso y con su participación y aunque se realizaba el algoritmo, el

hecho de que fuera texto dificultaba la comprensión. La forma de participar era mediante el chat, hablando, abriendo el micrófono o compartiendo su pantalla. Esta forma de trabajar resultó complicada para los alumnos, sobre todo para los que cursaban por primera vez una asignatura de programación. Alrededor del 25% reconocieron quedarse con dudas y entre las razones, se encontraban que se distraían y cuando querían retomar la clase ya era complicado y ni siquiera sabían qué preguntar, otros mencionaron que no preguntaban porque sabían que la clase se estaba grabando, pero mencionaban que las dudas de sus compañeros se habían atendido.

Para el segundo curso que se impartió ya con los antecedentes anteriores, se vio la necesidad de usar un pizarrón compartido en el que los estudiantes pudieran escribir de manera colaborativa. Se analizaron dos pizarras compartidas: NotebookCast y Jamboard, estas se usaron con los grupos y como resultado de la encuesta, los estudiantes mencionaron que realizar la codificación les era más fácil después de que el problema era analizado usando gráficos simples y podían participar de forma colaborativa en el diseño, además de poder descargar la pizarra y hacer apuntes del curso. En un inicio ambas pizarras presentaban la misma funcionalidad, pero en el momento en que Jamboard se podía iniciar directamente en la reunión de Meet, el invitar a colaborar para el Profesor era más simple. En la Figura 5 se muestra un ejemplo del uso de una pizarra.

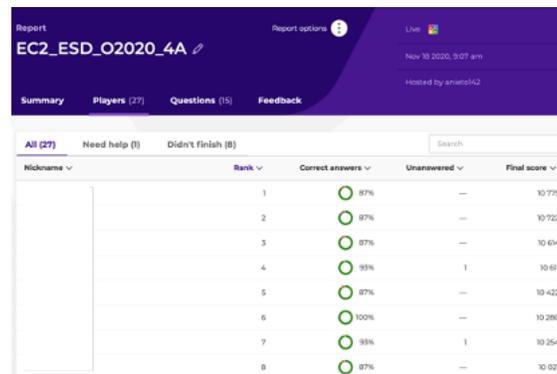


**Figura 5.** *Uso de una pizarra de NotebookCast. En la izquierda sólo la pizarra y en la derecha se muestra el uso de funciones apoyándose en el ejemplo de la pizarra*

Para codificación y depuración se usó Codelite y un compilador en línea para los que no podían instalar en su dispositivo. El proceso de codificación era iniciado por el profesor y los estudiantes participaban respondiendo preguntas y con apoyo del pizarrón se guiaba la implementación, realizaban de manera independiente pequeñas variantes o terminaban la implementación de una función en particular y se resolvían dudas en la reunión de Meet y se apoyaba de igual manera a encontrar errores, pero los estudiantes tenían que compartir su pantalla.

Por lo que respecta a la evaluación, en este segundo curso, las evidencias de conocimiento se realizaron usando Kahoot y de los dos grupos que se tenían en uno en particular el desempeño mejoró con respecto al uso de un formulario, en particular les gustaba ver cómo subían de posición. Sin embargo, en el otro grupo (que se habían detectado copias y respuestas que no pertenecían a la versión cuando se aplicaba formulario) mencionaron que los estresaba el

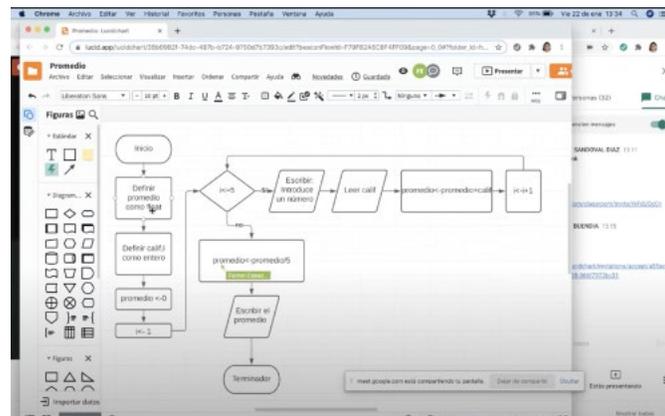
tener un tiempo para responder y los alumnos que habían copiado tuvieron un desempeño menor que en los formularios. En la Figura 6 se muestra un reporte de la aplicación de una evidencia de conocimiento.



Rank	Correct answers	Unanswered	Final score
1	87%	—	10 775
2	87%	—	10 722
3	87%	—	10 694
4	95%	1	10 681
5	87%	—	10 422
6	100%	—	10 280
7	98%	1	10 254
8	87%	—	10 021

**Figura 6.** Reporte de Kahoot de una aplicación de una evidencia de conocimiento

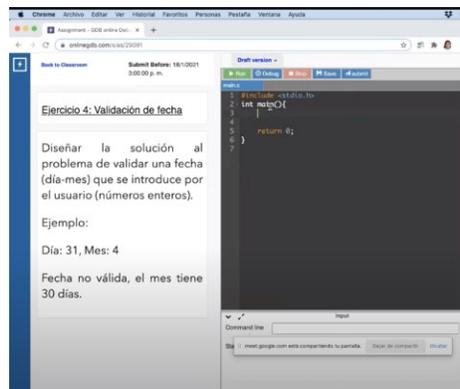
Durante la impartición de estos cursos se estableció que el principal problema es el diseño de la solución, aún más que la implementación, por lo cual, al impartir la asignatura de Programación Estructurada para estudiantes de Ingeniería en Electrónica y Telecomunicaciones y después de analizar los resultados de la evaluación diagnóstica, se procedió a buscar una herramienta para elaborar el diseño y que pudiera ser colaborativa. Se seleccionó Lucidchart que, aunque no es gratuita, la funcionalidad que ofrece de forma gratuita sería suficiente para el uso que se requiere. En la Figura 7 se muestra el desarrollo de un diagrama de flujo, los estudiantes pueden participar en la elaboración y se comparte al grupo.



**Figura 7.** Uso de Lucidchart para realizar un diagrama de flujo de forma colaborativa.

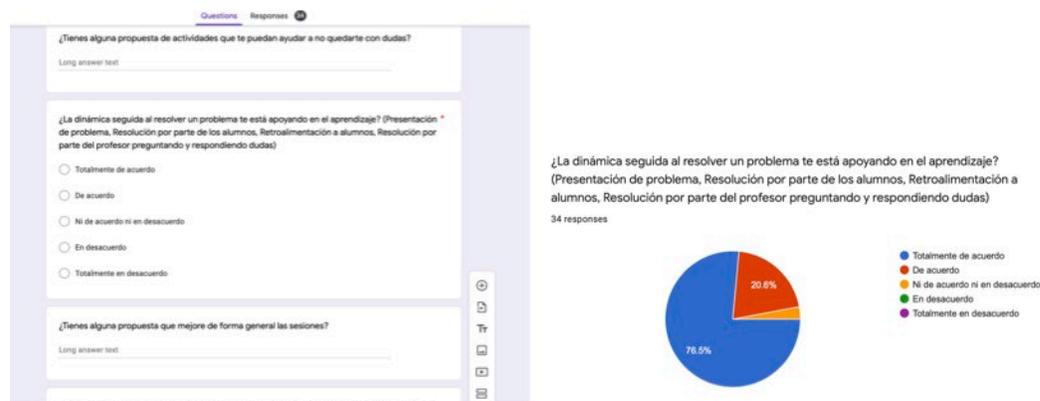
A este último curso se agregó también el uso de un compilador en línea, en el cual los estudiantes pudieran realizar las actividades y el apoyar y dar retroalimentación a la actividad sea de manera más rápida que el que ellos compartan pantalla. El compilador seleccionado es OnlineGDB, en el cual se creó un Classroom (nueva funcionalidad) en el cual se asignan actividades y los alumnos pueden enviar. Este mismo compilador también cuenta con un

depurador que puede ser usado y se puede dar retroalimentación en comentarios. En la Figura 8 se muestra un ejemplo de una asignación y de cómo ellos pueden enviar la actividad asignada.



**Figura 8.** *Uso de un Classroom en OnlineGDB*

En la Figura 9 se muestra una de las encuestas realizadas en la que el principal objetivo era medir el grado de satisfacción de los estudiantes y que también incluía preguntas abiertas para que pudieran comentar y dar retroalimentación, el 76.5% considera que la forma de resolver un problema y, por ende, las herramientas les está ayudando en el aprendizaje.



**Figura 9.** *Encuesta aplicada a los estudiantes*

Actualmente, se están revisando compiladores que permitan trabajar de forma colaborativa y permitan al profesor evaluar el trabajo en equipo, esto debido a que en la encuesta los estudiantes mencionan que lo que les falta es tener más supervisión al momento de trabajar en actividades no sumativas, el saber que el profesor no puede ver lo que están haciendo

## CONCLUSIONES

La situación actual impidió que la educación presencial continuara como se venía realizando, eso cambió la forma de impartir las clases y la educación a distancia requiere el diseño de material diferente y usar herramientas que permitan al estudiante aprender haciendo. Los estudiantes de la Universidad nunca habían tenido clase de forma no presencial, ni siquiera

de forma semi-presencial y esto representó un gran reto para los profesores y para los mismos estudiantes.

Particularmente, en una asignatura de programación se requiere que el alumno participe de forma activa, realice ejercicios desde el análisis del problema hasta la depuración y pruebas, requiere hacer para poder desarrollar la lógica es por eso muy importante que se encuentren herramientas que apoyen a que el estudiante aprenda haciendo y que gracias a la forma sincrónica el profesor pueda ayudar a resolver dudas, porque si bien en el modelo de educación basado en competencias el alumno es el actor principal y responsable de su aprendizaje en esta modalidad su compromiso debe ser mayor, pero requiere el apoyo de herramientas.

Las herramientas empleadas son: Google Classroom, Google Meet, Formularios, Kahoot, Jamboard, Lucidchart y OnlineGDB. El compilador en línea apoya a que el estudiante tenga el acompañamiento del profesor durante la reunión de Meet. Fuera de las sesiones de clase el estudiante cuenta con las grabaciones de las sesiones, las diapositivas, los diagramas y los códigos, sin embargo, de acuerdo con Mora, Rodríguez y García (2017), el material didáctico para educación a distancia debe cumplir con ciertas características y se está trabajando para que el material generado las cumpla. Otro trabajo futuro es el incorporar herramientas que permitan que los estudiantes trabajen en equipo al realizar actividades de programación y que el profesor pueda evaluar la participación de cada integrante del equipo.

## BIBLIOGRAFÍA

Biggs, J. (2004). *Calidad del Aprendizaje Universitario*. Madrid: Narcea.

Mora, R., Rodríguez, N. y García, L. (2017). Estructura de materiales didácticos para la educación virtual que faciliten la comprensión de contenidos temáticos. *Revista ANFEI Digital*. <https://anfei.mx/revista/index.php/revista/article/view/350>

Mujica, R. (23 de septiembre de 2020). Beneficios del aprendizaje sincrónico. [blog]. Docentes 2.0. <https://blog.docentes20.com/2020/09/beneficios-del-aprendizaje-sincronico-docentes-2-0/>

Sánchez, J. (2003). Integración Curricular de TICs Concepto y Modelos. *Revista Enfoques Educativos*, 5(1), 51-56. <https://enfoqueseducacionales.uchile.cl/index.php/REE/article/view/47512>

Santoveña, S. (2012). El proceso de enseñanza-aprendizaje a través de herramientas de comunicación síncrona: El caso de Elluminate Live. *Electronic Journal of Research in Educational Psychology*, 10(1), 447-474. <https://www.redalyc.org/pdf/2931/293123551022.pdf>