

**“EFECTOS DEL USO DE OPEN EDUCATIONAL RESOURCES
EN LA CREACIÓN DE RETROALIMENTACIÓN FORMATIVA,
UTILIZANDO UNA HERRAMIENTA DE CORRECCIÓN
ELECTRÓNICA Y SU IMPACTO EN EL APRENDIZAJE DEL
ESTUDIANTE”**

Hans Cristian Jeria Díaz

Universidad Adolfo Ibáñez

Facultad de Ingeniería y Ciencias

Santiago de Chile

Diciembre de 2017

EFFECTOS DEL USO DE OPEN EDUCATIONAL RESOURCES EN LA CREACIÓN DE RETROALIMENTACIÓN FORMATIVA, UTILIZANDO UNA HERRAMIENTA DE CORRECCIÓN ELECTRÓNICA Y SU IMPACTO EN EL APRENDIZAJE DEL ESTUDIANTE

Hans Cristian Jeria Diaz

**TESIS PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL MENCIÓN TECNOLOGÍAS DE
LA INFORMACIÓN Y AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, TECNOLOGÍAS
DE LA INFORMACIÓN**

Director Académico MCI-TI:

Miguel Carrasco

Profesor guía:

Jorge Villalón

Línea de Investigación:

Tecnología y Educación

Universidad Adolfo Ibáñez

Facultad de Ingeniería y Ciencias

Santiago de Chile

Diciembre de 2017

“Si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes”

Isaac Newton

“La educación es el arma más poderosa que puedes usar para cambiar el mundo.”

Nelson Mandela

El trabajo del maestro no consiste tanto en enseñar todo lo aprendible, como en producir en el alumno amor y estima por el conocimiento.

John Locke

Agradecimientos

A todas las personas que participaron y/o aportaron a la realización de esta investigación, muchas gracias:

Jorge Villalón, profesor guía y gran motivador para llevar a cabo esta investigación,

Tamara Figueroa, por su apoyo incondicional de principio a fin,

Dina Díaz, Cristian Jeria y Dana Jeria, por el apoyo a través de mi formación profesional,

Rafael Cereceda, profesor de Programación y Sistemas de información por su apoyo,

Claudio Díaz, profesor de Sistemas de Información por su apoyo,

Miguel Carrasco, profesor de Programación por su apoyo,

José Antonio Morales, corrector experimental curso Sistemas de Información,

Renzo Stefano Tassara Miller, corrector experimental curso Sistemas de Información,

Belén Céspedes Stiven, corrector experimental curso Programación,

Nicolás Pizarro, corrector experimental curso Programación,

José Contreras, corrector experimental curso Programación,

Daniel Cortes, corrector experimental curso Programación,

Karim Donoso, Coordinadora plataforma Webcursos,

Mihail Pozarski, Jefe de Desarrollo Webcursos.

Resumen

Se realizó una investigación para determinar los impactos sobre el rendimiento académico de los estudiantes, utilizando los recursos denominados “*Open Educational Resources*” como herramienta de retroalimentación para el evaluador educacional (o corrector). La metodología implica la medición del rendimiento del estudiante entre la primera evaluación y la nota final obtenida en el curso sujeto de investigación. Para realizar dicho proceso los evaluadores (o correctores) utilizarán la herramienta *Emarking*, que permite la corrección online de evaluaciones tomadas en papel.

Los experimentos se llevarán a cabo en dos cursos diferentes, en primer lugar, el curso de Programación de primer semestre de la carrera de Ingeniería Civil de la Universidad Adolfo Ibáñez y, en segundo lugar, el curso de Sistemas de Información del noveno semestre de la carrera de Ingeniería Civil de la Universidad Adolfo Ibáñez.

A partir de los datos obtenidos, se generarán modelos estadísticos para analizar el comportamiento de la herramienta y los nuevos componentes a través de los cursos descritos, con el fin de obtener resultados estadísticamente relevantes para la investigación.

Palabras clave: corrección electrónica, Emarking, open educational resources, retroalimentación.

Abstract

Research was conducted to determine the impacts on student's academic performance, using resources called "Open Educational Resources" as a feedback tool for the educational evaluator (or marker). The methodology involves measuring the student's performance between the first evaluation and the final grade obtained in the subject course of research. To carry out this process, the evaluators (or markers) will use the Emarking tool, which allows the online correction of evaluations taken on paper.

The experiments will be carried out in two different courses, firstly, the course of Programming of the first semester of the career of Civil Engineering of the Adolfo Ibáñez University and, secondly, the course of Information Systems of the ninth semester of the Civil Engineering degree from the Adolfo Ibáñez University.

Based on the data obtained, statistical models will be generated to analyze the behavior of the tool and the new components through the courses described, in order to obtain statistically relevant results for the investigation.

Keywords: online marking, Emarking, open educational resources, feedback.

Índice de contenidos

CAPÍTULO 1: CONTEXTO	22
1.1 ESCENARIO	22
1.2 CORRECCIÓN MANUAL Y CREACIÓN DE RETROALIMENTACIÓN	23
1.3 CORRECCIÓN EMARKING Y CREACIÓN RETROALIMENTACIÓN	23
1.4 OPORTUNIDAD	25
1.5 ELECCIÓN DE CURSOS CONTROL Y EXPERIMENTAL	25
CAPÍTULO 2: MARCO TEÓRICO.....	27
2.1 DEFINICIÓN RETROALIMENTACIÓN FORMATIVA	27
2.2 CALIDAD EN RETROALIMENTACIÓN FORMATIVA.....	27
2.3 MARCO DE TRABAJO PARA RETROALIMENTACIÓN FORMATIVA	29
2.4 HERRAMIENTA EMARKING Y NUEVOS DESARROLLOS	30
2.4.1 <i>Nuevas características para el profesor</i>	31
2.4.2 <i>Nuevas características para el corrector</i>	32
2.4.3 <i>Nuevas características para el estudiante</i>	34
2.5 GENERACIÓN DE LOS DATOS	35
2.6 MÉTODOS PARA ADQUISICIÓN DE DATOS	35
2.7 MÉTODOS PARA EL ANÁLISIS DE DATOS.....	35
CAPÍTULO 3: DISEÑO EXPERIMENTAL.....	36
3.1 PREGUNTA A RESPONDER	36
3.2 HIPÓTESIS	36
3.3 SELECCIÓN DE VARIABLES	37
3.3.1 <i>Variables de entrada</i>	38
3.3.2 <i>Variables de salida</i>	39
3.4 RESTRICCIONES Y FACTIBILIDAD DEL EXPERIMENTO	39
3.5 SELECCIÓN DEL MODELO EXPERIMENTAL	40
CAPÍTULO 4: METODOLOGÍA.....	41
4.1 FUNCIONALIDADES NUEVAS PARA EMARKING	41
4.1.1 <i>Diseño</i>	41
4.1.2 <i>Desarrollo</i>	43
4.2 ENTRENAMIENTO DE CORRECTORES, GRUPOS CONTROL Y EXPERIMENTAL	43
4.3 COORDINACIÓN DE EXPERIMENTO	44
4.4 ADQUISICIÓN DE DATOS.....	45
CAPÍTULO 5: ANÁLISIS DE RESULTADOS.....	46
5.1 ACONTECIMIENTOS RELEVANTES	46
5.2 ANÁLISIS DEL CURSO DE PROGRAMACIÓN.....	47
5.2.1 <i>Variables, normalización y supuestos del modelo MANOVA</i>	48
5.2.2 <i>Modelo MANOVA y análisis de resultados</i>	52
5.2.3 <i>Conclusiones</i>	58
5.3 ANÁLISIS CURSO DE SISTEMAS DE INFORMACIÓN	59

5.3.1	<i>Variables, normalización y supuestos del modelo MANOVA</i>	59
5.3.2	<i>Modelo MANOVA y análisis de resultados</i>	62
5.3.3	<i>Conclusiones</i>	70
DISCUSIÓN		72
CONCLUSIÓN Y RECOMENDACIONES		74
REFERENCIAS		78
ANEXOS		81
A	PAPER PUBLICADO EN CONFERENCIA EN EL MARCO DEL MCI	81
B	MARCO DE TRABAJO PROPUESTO POR VALERIE SHUTE	86
C	PRUEBAS DE PROGRAMACIÓN Y SISTEMAS DE INFORMACIÓN	90
C.1	PRUEBAS 1 PROGRAMACIÓN	90
C.2	RÚBRICA PRUEBAS 1 PROGRAMACIÓN	106
C.3	PRUEBAS 2 PROGRAMACIÓN	110
C.4	RÚBRICA PRUEBAS 2 PROGRAMACIÓN	118
C.5	EXAMEN PROGRAMACIÓN	119
C.6	RÚBRICA EXAMEN PROGRAMACIÓN	125
C.7	PRUEBAS 1 SISTEMAS DE INFORMACIÓN	127
C.8	RÚBRICA PRUEBAS 1 SISTEMAS DE INFORMACIÓN	132
C.9	PRUEBAS 2 SISTEMAS DE INFORMACIÓN	133
C.10	RÚBRICA PRUEBAS 2 SISTEMAS DE INFORMACIÓN	138
C.11	PRUEBAS 3 SISTEMAS DE INFORMACIÓN	139
C.12	RÚBRICA PRUEBAS 3 SISTEMAS DE INFORMACIÓN	139
D	SCRIPT EXTRACCIÓN DE DATOS DESDE BASE DE DATOS	140
F	PRESENTACIÓN DE CAPACITACIÓN CORRECTORES	148

Lista de figuras

FIGURA 1. CALIFICACIÓN DE CRITERIOS DE LA RÚBRICA EN EMARKING NORMAL.....	24
FIGURA 2. FORMULARIO CREACIÓN/EDICIÓN EMARKING CON FOCO EN PALABRAS CLAVES Y REPOSITARIOS O FUENTES DE OER.	32
FIGURA 3. WIDGET DE CORRECCIÓN NORMAL DE EMARKING.	33
FIGURA 4. WIDGET CREADO PARA LA REALIZACIÓN DEL EXPERIMENTO EN EMARKING O TAMBIÉN LLAMADO. ...	33
FIGURA 5. PRUEBA DESDE LA VISTA DEL ALUMNO, REVISIÓN DE RETROALIMENTACIÓN.	34
FIGURA 6. DIAGRAMA DE CAJA NEGRA.....	37
FIGURA 7. DIAGRAMA DE INTERACCIÓN CORRECTOR – OER.	42
FIGURA 8. MODELO MANOVA.....	52
FIGURA 9. MODELO MANOVA.....	62

Lista de gráficos

GRÁFICO 1. HISTOGRAMA DE FRECUENCIAS PARA NOTA DE PRUEBA 1 POR TIPO DE EVALUACIÓN.	53
GRÁFICO 2. HISTOGRAMA DE SUMA TOTAL DE TIEMPO DE REVISIÓN PRUEBA 1 POR RANGO DE NOTA OBTENIDA Y POR TIPO DE EVALUACIÓN. TIEMPO MEDIDO EN MINUTOS.	54
GRÁFICO 3. HISTOGRAMA DE FRECUENCIA DE NOTA FINAL POR TIPO DE EVALUACIÓN.	56
GRÁFICO 4. HISTOGRAMA DIFERENCIAS PROMEDIO ENTRE NOTA FINAL Y PRUEBA 2 RESPECTO A LA NOTA FINAL OBTENIDA POR EL ALUMNO Y POR TIPO DE EVALUACIÓN.	57
GRÁFICO 5. HISTOGRAMA DE FRECUENCIAS DE NOTA PRUEBA 3 POR TIPO DE EVALUACIÓN.	63
GRÁFICO 6. HISTOGRAMA DE TIEMPO TOTAL DE REVISIÓN DE NOTA PRUEBA 3 POR TIPO DE EVALUACIÓN. TIEMPO MEDIDO EN MINUTOS.	64
GRÁFICO 7. HISTOGRAMA DE FRECUENCIA DE NOTA PRUEBA 3 POR TIPO DE EVALUACIÓN.	66
GRÁFICO 8. HISTOGRAMA DE TIEMPOS TOTALES DE REVISIÓN DE NOTA PRUEBA 1, 2 Y 3 POR CURSO EXPERIMENTAL. TIEMPO MEDIDO EN MINUTOS.	67
GRÁFICO 9. HISTOGRAMA DE TIEMPOS TOTALES DE REVISIÓN DE NOTA PRUEBA 1, 2 Y 3 POR CURSO CONTROL. TIEMPO MEDIDO EN MINUTOS.	68
GRÁFICO 10. HISTOGRAMA DE DIFERENCIA PROMEDIO ENTRE NOTA FINAL Y PRUEBA 1 POR TIPO DE EVALUACIÓN.	69

Lista de tablas

TABLA 1. COMBINACIONES TIPO DE SECCIÓN Y TIPO DE CORRECCIÓN	38
TABLA 2. FECHAS DE EVALUACIONES PROGRAMACIÓN Y SISTEMAS DE INFORMACIÓN.	44
TABLA 3. ESTADÍSTICOS DESCRIPTIVOS. TIEMPO MEDIDO EN SEGUNDOS.	49
TABLA 4. PRUEBA DE NORMALIDAD KOLMOGOROV-SMIRNOV.	50
TABLA 5. PRUEBA BOX.	51
TABLA 6. PRUEBA LEVENE.	51
TABLA 7. MODELO MULTIVARIANTE.	52
TABLA 8. PRUEBAS DE EFECTO INTER-SUJETO ANOVA.	52
TABLA 9. PRUEBAS T-STUDENT PARA NOTA PRUEBA 1.	54
TABLA 10. PRUEBAS T-STUDENT PARA NOTA FINAL.	56
TABLA 11. ESTADÍSTICOS DESCRIPTIVOS. TIEMPOS MEDIDOS EN SEGUNDOS.	60
TABLA 12. PRUEBA DE NORMALIDAD KOLMOGOROV-SMIRNOV.	61
TABLA 13. PRUEBA BOX.	61
TABLA 14. PRUEBA LEVENE.	61
TABLA 15. MODELO MULTIVARIANTE.	62
TABLA 16. PRUEBAS DE EFECTO INTER-SUJETO ANOVA.	63
TABLA 17. PRUEBAS T-STUDENT PARA NOTA PRUEBA 3.	65
TABLA 18. PRUEBAS T-STUDENT PARA NOTA FINAL.	67

Introducción

En Chile el uso de la tecnología en las evaluaciones, en el contexto educativo, es aún bajo comparado con la utilización del lápiz y el papel. Del mismo modo, la retroalimentación presente en las evaluaciones es escasa o prácticamente nula, principalmente por la carga extra de trabajo que aquello significa tanto para el profesor o el corrector designado.

El uso de retroalimentación en el contexto educacional es considerada crucial para mejorar el conocimiento y la adquisición de habilidades. Shute define la retroalimentación formativa como: *“... información comunicada hacia el aprendiz con la intención de modificar su pensamiento o comportamiento con el propósito de mejorar el aprendizaje”* [19]. La retroalimentación formativa, intencional o no intencional, en los primeros años de la educación superior es particularmente crucial para los estudiantes, no solo cumple un rol vital en su aprendizaje, sino también influye en su identidad profesional [6]. De este modo, el uso de retroalimentación formativa puede convertir una evaluación en un instrumento de desarrollo del aprendizaje [8], incluso de manera significativa, la retroalimentación puede mejorar la profundidad del aprendizaje [12].

A pesar de los beneficios y calidad reconocida que la retroalimentación formativa tiene, aún existen múltiples desafíos para asegurar la calidad de la retroalimentación, como su construcción que requiere un esfuerzo importante. Estudios sobre aquello, exploran la calidad de la retroalimentación desde el punto de vista del estudiante, los cuales encontraron que la gran mayoría considera que la retroalimentación es difícil de entender, incluyendo comentarios impersonales o estandarizados, los cuales son percibidos de forma negativa y con falta de información sobre la respuesta del

estudiante [8]. Tales hallazgos están en línea con Boud y Falchicov, quienes identificaron que factores personales e institucionales afectan la forma en la que los estudiantes experimentan y consumen la retroalimentación [4], esto significa que no es posible encontrar un estándar común en el escenario de la educación superior. Prince propone que el potencial de la retroalimentación solo puede ser aprovechado si el estudiante es entrenado en cómo interpretar la retroalimentación, para realizar las conexiones entre el trabajo realizado, la correspondiente retroalimentación y las formas por las cuales podría mejorar a futuro. [18] Sin embargo, Evans plantea preocupaciones sobre la necesidad de entrenar a los estudiantes sobre cómo interpretar la retroalimentación, se debe tener cuidado con proveer directivas o guías muy explícitas en la interpretación de la retroalimentación, ya que, existe el peligro inherente de orientar explícitamente al estudiante lo que puede generar dependencia y pensamiento limitado. [7]

Valerie Shute [20], como síntesis de su investigación, que abarca las publicaciones de los últimos treinta años respecto a la retroalimentación, plantea un marco de trabajo que se enfoca en qué cosas hacer, qué cosas evitar, incidentes respecto al tiempo y características del alumno. **(Anexo B)**

El uso de tecnología que aporte al proceso de generación de la retroalimentación formativa puede ayudar a atacar el problema de los crecientes costos para el profesor y el corrector. La tecnología puede ayudar a reducir dichos costos permitiendo que la creación de la retroalimentación sea colaborativa, reutilizando retroalimentación dentro de evaluaciones y simplificando su entrega hacia los estudiantes. Sin embargo, todas estas características son de un dominio independiente, y no pueden beneficiarse de una fuente en específico, para ayudar a los estudiantes a expandir su conocimiento base, como Open Educational Resources (OER).

Seymour Papert propuso en Mindstorms que es posible modularizar el conocimiento a través de la tecnología, el conocimiento puede ser partido en trozos que el llamo *mind-sizes* lo que permitirá que sea más comunicable, digerible y fácil de construir [16]. A partir de las ideas de Papert, presentadas hace más de treinta y cinco años, ahora es posible encontrar dicha modularidad en el formato que se conoce como Open Educational Resources, que no tiene restricciones territoriales o económicas. La modularidad de los OER proviene de la necesidad de su reutilización, por lo tanto se convierten en excelentes fuentes para mejorar la retroalimentación, al ser explicaciones autónomas de dominios específicos que facilitan su comprensión y, al conectarse a otros recursos y por lo tanto abrir una ventana a los estudiantes a un fuente de conocimiento del dominio para su propia reflexión.

El número de OER disponibles ha aumentado progresivamente en los últimos años y, según la *Organization for Economic Cooperation and Development* (OECD), hay más de 3.000 cursos de acceso abierto y más de 300 universidades de todo el mundo que comparten contenido [22].

De este modo, proponemos que OER pueden ser una excelente fuente de información de calidad en la retroalimentación formativa, y dicha tecnología puede facilitar su uso dentro del proceso de evaluación, ayudando a crear conciencia de tales repositorios dentro de profesores, correctores y estudiantes.

En estudios previos, se informó sobre la herramienta Emarking [20], desarrollada en la Universidad Adolfo Ibáñez, y sujeto de múltiples paper investigativos en el ámbito de la aplicación de Tecnologías de la Información en la Educación. Es por ello, que la propuesta de usar OER para la generación de la retroalimentación se llevará a cabo de la mano de la herramienta Emarking que se usa para corregir de forma online pruebas tomadas en papel y, de este modo, es un aliado estratégico para llevar a cabo los experimentos necesarios para la investigación.

Este estudio propone evaluar el uso de OER en la creación de retroalimentación formativa, realizando la corrección de la evaluación y la creación de la retroalimentación de manera online en el software Emarking, con el fin de medir el impacto producido en el rendimiento del estudiante.

Capítulo 1: Contexto

En este capítulo se presenta el contexto de la investigación, haciendo referencia a la institución donde se realizó el experimento, descripción de los procesos involucrados y los cursos que participaron de la investigación.

1.1 Escenario

La Universidad Adolfo Ibáñez usa como uno de sus principales canales de comunicación entre la institución y sus alumnos la plataforma Webcursos. Esta es una aplicación basada en el software de código abierto Moodle, el cual a su vez es un sistema de gestión del aprendizaje o LMS (Learning Management System) por sus siglas en inglés. Ofrece múltiples características, tanto para coordinadores de secciones, profesores, ayudantes y estudiantes. Entre sus principales funciones se encuentran la entrega y almacenamiento de trabajos, asignación de calificaciones, toma de asistencia y corrección online de evaluaciones tomadas en papel.

Desde el año 2012, se incorporó a Webcursos un plugin desarrollado por un equipo de profesionales de la Facultad de Ciencias e Ingeniería, llamado Emarking, basado en código abierto, donde su principal función es agregar valor una vez que la prueba fue tomada en papel por los estudiantes, permitiendo realizar todo el proceso posterior de forma online, es decir; corrección, entrega de notas y corrección se realiza online e independiente de la prueba original en papel.

Actualmente, el uso de Emarking en la universidad es extenso, ya no solo es utilizado por cursos del área de tecnologías de la información, sino también, por otros cursos tales como: filosofía, política, expresión escrita, finanzas, cálculo, optimización,

investigación de operaciones y sistemas de información entre otros. Es por ello, que se considera a Emarking un software de estado maduro a nivel de usabilidad y de características, que ha ido incrementando su porcentaje de uso en la universidad a un ritmo creciente y adecuándose para cumplir los requerimientos específicos internos.

1.2 Corrección manual y creación de retroalimentación

En Chile es aún bastante utilizado en la corrección de evaluaciones, el uso de papel y el proceso manual, principalmente en preguntas de estilo abierto. El proceso general de corrección manual de la Universidad Adolfo Ibáñez inicia con la impresión de pruebas no indexadas, es decir, se imprime la cantidad de pruebas igual al número de estudiantes correspondiente a la lista del curso, que a su vez no identifica a cada alumno. Posteriormente, los alumnos contestan y las pruebas son recolectadas y entregadas a los correctores, luego las evaluaciones son corregidas y las notas publicadas online y, las pruebas originales retornadas a los estudiantes, para que estos últimos revisen y puedan realizar el proceso de recorreción si lo desean.

Cabe destacar, que el proceso de corrección y creación de la retroalimentación, bajo esta modalidad es cooperativa, es decir, se suman las partes realizadas por dos o más correctores, en desmedro de un proceso colaborativo que es capaz de entregar un mayor valor en el contexto educativo.

1.3 Corrección Emarking y creación retroalimentación

El proceso con la herramienta Emarking contempla los subprocesos de envío de prueba a impresión, digitalización, corrección en pantalla de las evaluaciones tomadas en papel y finalmente la publicación de pruebas y recorreción. Como se mencionó previamente, Emarking es un software con 5 años de antigüedad, al ser un proyecto de innovación y desarrollo se han realizado múltiples estudios a su alrededor, entre los cuales se

encuentra el análisis del efecto de grupos de correctores y su colaboración en la corrección [13], costo y beneficio del método Delphi aplicado al entrenamiento de correctores [21] o presentación de Emarking como una herramienta para evaluaciones tomadas en papel. [20] Este último estudio, presenta resultados concluyentes que indican que no existen diferencias entre corregir en papel y Emarking a nivel de notas finales, es decir, no se puede asumir diferencias si se usa un método por sobre otro para calificar las pruebas.

La **Figura 1** muestra la forma normal mediante la cual se califica cada criterio de la rúbrica de la evaluación, además, se observan las herramientas disponibles para la creación de retroalimentación, esto ocurre para cada criterio de la rúbrica. Las herramientas son: el nivel de la rúbrica que asocia un puntaje establecido, un comentario optativo por parte del ayudante y un ajuste de puntaje.

The screenshot displays the Emarking Normal interface. At the top, a student's information is shown: NOMBRE: ALVAREZ SAYES, DILAN ALEJANDRO; RUT: 18862534; CURSO: SISTEMAS DE INFORMACIÓN SEC.1 STGO FIC STGO 2017/1; PAGINA: 3 DE 5. A QR code is visible next to the student's name. The main area shows a student's handwritten response to a problem. Overlaid on this is a modal window titled 'Agregar/Editar corrección'. The modal contains a dropdown menu for 'Nivel' with the selected option 'Detecta el problema u oportunidad y describe en detalle (4)'. Below this is a text area for 'Comentario'. At the bottom of the modal is a 'Ajustar puntaje' field with a '+0' button. To the right of the modal, a table lists criteria (P1, P2, P3) with their respective scores (4/4, 6/6, 2/6) and descriptions. At the bottom right, there is a 'Comentarios' section with a list of comments.

Criterio	Puntaje	Comentario
P1	4 / 4	Hace una un proble clara
P2	6 / 6	Respuesta responde
P3	2 / 6	Respuesta responde

FIGURA 1. CALIFICACIÓN DE CRITERIOS DE LA RÚBRICA EN EMARKING NORMAL.

1.4 Oportunidad

Como se comentó en secciones previas, alrededor de la herramienta Emarking y sus cinco años de uso en la universidad, numerosos estudios se han llevado a cabo, pero el tópico de la retroalimentación, tanto en creación e impacto en el aprendizaje del estudiante, nunca fue explorado. De este modo, nace la pregunta ¿podrá una herramienta ayudar a alentar al corrector a crear retroalimentación?, de la mano nace la segunda pregunta ¿cuál será el impacto en el aprendizaje del estudiante; al disponer de retroalimentación creada por el corrector usando el nuevo widget con OER?

Una investigación previa [10] (**Anexo A**) a la presentada en este documento, realizada dentro del magíster en ciencias, tuvo como punto de partida la primera pregunta de las mencionadas previamente. Los resultados fueron reportados en la conferencia IEEE International Conference of Advanced Learning Technologies (ICALT) 2017. En términos generales, el largo de la retroalimentación creada por los correctores es afectado directamente por la presencia del nuevo widget, que consume contenido OER y genera enlaces hacia ellos. Solo el grupo experimental tuvo acceso al widget.

En consecuencia, el presente documento de investigación está basado en la pregunta ¿cuál será el impacto en el aprendizaje del estudiante, al disponer de retroalimentación creada por el corrector usando el nuevo widget con conexión hacia OER?

1.5 Elección de cursos control y experimental

En la investigación, resultó interesante observar que podrían existir diferencias entre cursos de primer año y último año de educación superior en su comportamiento respecto a la herramienta Emarking. De esta manera, se utilizaron cuatro secciones del curso de Programación de primer semestre, primer año, de la carrera Ingeniería Civil,

con dos secciones control y dos secciones experimentales. También, se utilizaron tres secciones del ramo de Sistemas de Información de noveno semestre, último año, de Ingeniería Civil, con dos secciones control y una sección experimental.

Se utilizaron, en Programación dos pruebas llevadas a cabo durante el semestre, más el examen final del ramo. En Sistemas de Información tres pruebas durante el semestre, en este caso el examen correspondía a un trabajo grupal semestral. En **Anexo C**, se pueden encontrar las seis evaluaciones antes mencionadas, con sus respectivas rubricas utilizadas para la corrección.

Cabe destacar que, en ambos cursos durante la primera evaluación todos utilizaron Emarking sin ninguna modificación, es decir sin el widget con OER. Aquello para tener una base de comparación, debido a que interesa explorar el nivel de aprendizaje de los estudiantes más que comparar la sección experimental con control, es comparar al estudiante consigo mismo.

En secciones posteriores, se mencionará la cantidad de alumnos que participaron del experimento, tanto por cursos y secciones control y experimental.

Capítulo 2: Marco teórico

En este capítulo se definen y describen los conceptos y herramientas utilizadas en la investigación. El orden corresponde cronológicamente con el avance de la investigación, desde definiciones y exploración de la literatura, luego desarrollo de software en la herramienta Emarking, siguiendo con generación y adquisición de datos y finalizando con los métodos para el análisis de los datos.

2.1 Definición retroalimentación formativa

De acuerdo a Cohen [5], la retroalimentación es una de las características más poderosas e incomprensidas en el diseño educacional. Es por ello que, durante los últimos 30 años los estudios sobre este tópico han proliferado. La investigación de Valerie Shute, Focus on Formative Feedback [19], realiza un largo recorrido por la literatura abarcando el tema de la retroalimentación a cabalidad y postula la siguiente definición: *la retroalimentación formativa representa la comunicación de información hacia el estudiante, la cual, tiene la intención de modificar el pensamiento o comportamiento del estudiante con el propósito de mejorar el aprendizaje.*

2.2 Calidad en retroalimentación formativa

La investigación de David Nicole y Debra Macfarlane-Dick [15] presenta siete principios para la buena práctica de la retroalimentación, el contexto de su estudio es la auto regulación, pero es posible extrapolar sus principios al concepto de retroalimentación en general, por ser una guía más que un manual estricto. De todos modos, luego de enunciar los principios se destacarán los que aplican para la actual investigación.

- I. Ayuda a aclarar qué es un buen desempeño (objetivos, criterios, estándares esperados).
- II. Facilita el desarrollo de auto regulación (reflexión) en el aprendizaje.
- III. Entrega información de alta calidad al estudiante sobre su aprendizaje.
- IV. Alienta al profesor y el diálogo sobre el aprendizaje.
- V. Alienta positivamente comportamientos de autoestima.
- VI. Provee oportunidades para cerrar la brecha entre el desempeño actual y el esperado.
- VII. Provee al profesor información que puede utilizar para mejorar la enseñanza.

Como se puede apreciar, el contexto de auto regulación no afecta en sí a los principios dada la forma en la cual se expresaron. De este modo, los principios que se usaron para la investigación y fueron transmitidos hacia los correctores para su formación son los principios I, III, y VI.

De acuerdo a los autores Albertson [1], Azeveco [2] y Narciss [14], el objetivo principal de la retroalimentación formativa, ya sea que lo entregue un profesor o una computadora, en la casa de clases o en cualquier otro lugar, es mejorar el aprendizaje y/o el rendimiento. En la herramienta propuesta, Emarking con el agregado de un widget que proporciona Open Educational Resources, hay un enfoque en dos principios de retroalimentación de calidad, que lo estudiantes informan como un problema: Proporcionar información de alta calidad sobre su aprendizaje, mientras se aclara qué es un buen desempeño. El objetivo es apoyar la creación de comentarios una vez que el docente un nivel de rendimiento en la respuesta del alumno, e intervenir en el momento en que se realiza el juicio proponiendo Open Educational Resources que pueden ayudar a los alumnos a desarrollar su desempeño en dicho dominio.

Según Black & William [3], la retroalimentación tiene dos funciones principales: directiva y facilitadora. La función directiva informa al estudiante sobre lo que tiene

que corregir o revisar. Esta función tiende a ser más específica que la función facilitadora, la cual proporciona comentarios y sugerencias para ayudar a guiar a los estudiantes hacia su propia revisión y conceptualización. Además, desde la perspectiva de la información, la retroalimentación y sus comentarios pueden proporcionar a los estudiantes dos tipos de información: verificación y elaboración [11]. La verificación se define como el simple juicio de si una respuesta es correcta o no y, la elaboración es el aspecto informativo del mensaje, que proporciona pistas para guiar al alumno hacia una respuesta correcta.

En el diseño de la herramienta o widget, la atención se centra en la función facilitadora, ya que, se integra al interior de Emarking, el cual ya es una herramienta de corrección online que incluye rúbricas para la calificación, lo que proporciona una base estructurada para los juicios sobre el trabajo evaluado. Desde la perspectiva de la información, el propósito es apoyar la elaboración, ya que la información adicional se entra más allá de la exactitud de la respuesta, y permite precisamente guiar al estudiante hacia un conocimiento más amplio.

2.3 Marco de trabajo para retroalimentación formativa

Como resultado de su investigación, Valerie Shute propuso recomendaciones y guías para la retroalimentación formativa. En ellas propone que cosas hacer para mejorar el aprendizaje, que cosas evitar para mejorar el aprendizaje, respecto a incidencias relacionadas con el tiempo y por último características del estudiante. En **Anexo B**, se puede consultar en detalle todo el marco de trabajo propuesto que incluye todos los casos.

El marco de acciones que Valerie propone, abarca casi todos los aspectos y circunstancias que ocurren en la educación superior. De esta manera, es necesario hacer explícitas las limitaciones del experimento con el fin de dar las guías iniciales de cómo se abordará. Todos los cursos, es decir, las 7 secciones, sumando programación y

sistemas de información, usaron Emarking para corregir las evaluaciones tomadas en papel, por lo cual, la retroalimentación creada es en formato escrito y, por otro lado, los correctores tienen un periodo máximo de 14 días de corrido, desde la toma de la evaluación para la entrega de la evaluación corregida, dicha entrega ocurre de forma online a través del LMS Webcursos.

Posterior a la primera evaluación, tanto para las secciones de programación como de sistemas de información, se llevó a cabo una charla para los correctores, tanto control como experimental, donde se les explicó los principios I, III, y VI (de la **sección 2.2**) sobre la buena práctica en retroalimentación y sobre el marco de trabajo de Valerie Shute, tomando las consideraciones de que la retroalimentación es retrasada y en formato escrito. Todo aquello, con el fin de mejorar la creación de retroalimentación que ellos realizan. Para el caso experimental, la única diferencia fue que experimentaron el funcionamiento del widget que entrega OER, que permite enriquecer la retroalimentación mientras se usa Emarking para corregir.

2.4 Herramienta Emarking y nuevos desarrollos

Emarking es un proyecto de ingeniería de software, el cual empezó el año 2012 a cargo del profesor Jorge Villalón, buscando la eficiencia al momento de corregir evaluaciones tomadas en papel e introducir la tecnología en proceso educativos.

El proyecto se desarrolló como un plugin para el LMS Moodle, específicamente un plugin de módulo de actividad. Los principales lenguajes utilizados son PHP, debido a que Moodle está programado en dicho lenguaje, y la interfaz de corrección que está desarrolla en Java utilizando el framework Google Web Tool Kit (GWT), el cual permite transformar el desarrollo realizado en lenguaje Java, de alto nivel de complejidad, hacia el lenguaje JavaScript, un lenguaje amigable, maleable y ligero para los navegadores actuales.

En síntesis, Emarking para su backend utiliza PHP, es decir, conexiones con base de datos, conexiones con Moodle y conexiones con servicios externos y, para su frontend utilizada JavaScript, es decir, interacciones entre interfaz de corrección y usuario.

Como se mencionó en secciones previas, en el proyecto Emarking nunca se había planteado abordar la problemática de la creación de retroalimentación, por lo cual, en las siguientes **secciones 2.4.1, 2.4.2 y 2.4.3** se describen los desarrollos realizados, tanto a nivel de backend y frontend, para los principales actores del estudio.

2.4.1 Nuevas características para el profesor

Para incorporar la posibilidad que el corrector interactúe con Open Educational Resources, es necesario que el profesor genere palabras claves o frases claves desde la rúbrica de la evaluación, con el propósito que desde dichas palabras claves se realice búsqueda de material en los repositorios elegidos.

Los repositorios elegidos para ser utilizados como fuente en el experimento fueron:

- Massachusetts Institute of Technology (MIT) OpenCourseWare¹.
- Multimedia Educational Resource for Learning and Online Teaching² (MERLOT).
- Webcursos³, plataforma Moodle de la Universidad. Material educación seleccionado y subido por ayudante y/o profesor.

Una vez creadas las palabras claves o frases claves, desde la interfaz de creación o edición de la actividad Emarking el profesor en el formulario, **Figura 2**, debe colocar separado por coma cada palabra o frase clave y a la vez seleccionar los repositorios que desea utilizar.

¹ OpenCourseWare MIT Webpage: <https://ocw.mit.edu>

² MERLOT Webpage: <https://www.merlot.org>

³ Webcursos Webpage: <https://webcursos.uai.cl>

PRUEBA 2

Imprimir y digitalizar Corrección en pantalla Reportes Configuración

Corrección Comentarios predefinidos Correctores Páginas Resultados Importar rúbrica Exportar

Estudiantes ven pruebas de otros estudiantes No

Corrección anónima Estudiante anónimo / Corrector visible

Preguntar percepción de justicia Deshabilitada

Palabras claves para Feedback
supply chain management, customer relationship management, e-commerce, enterprise content management, business analytics, enterprise resource planning

Nombre de las fuentes OER mit,merlot

FIGURA 2. FORMULARIO CREACIÓN/EDICIÓN EMARKING CON FOCO EN PALABRAS CLAVES Y REPOSITORIOS O FUENTES DE OER.

2.4.2 Nuevas características para el corrector

Como se menciona en la sección anterior, una vez creadas las palabras o frases claves, durante el proceso de corrección el widget para calificar las preguntas mediante los criterios de la rúbrica despliega contenido de los repositorios. La **Figura 3** muestra el widget en un proceso normal de Emarking y en la **Figura 4** se muestra el nuevo widget utilizado por los cursos experimentales.

Observando la **Figura 4**, se aprecian en el lado superior derecho las palabras o frases claves, las cuales al presionar realizan una búsqueda en los repositorios de material educativo. Posteriormente, el corrector puede revisar el contenido y la fuente de procedencia, en este momento, puede elegir qué contenido agregar como retroalimentación, la lista de material agregado se observa en el lado inferior izquierdo. Igual que en el widget normal de Emarking es posible escribir un comentario, elegir la calificación y ajustar el puntaje.

Agregar/Editar corrección
P2

Nivel: Hace referencia a los cuidados a tener considerando uno c

Comentario:

Ajustar puntaje: +0

Guardar Cancelar

FIGURA 3. WIDGET DE CORRECCIÓN NORMAL DE E-MARKING.

Agregar/Editar corrección
P3b

Nivel: Responde claramente sobre la creación de

Comentario: No sirve implementar un CMS si la empresa no cuenta con la información digitalizada por lo que se requiere hacer uso de un ECM en primer lugar. Se menciona brevemente E-commerce pero sin nombrar una estrategia específica.

Ajustar puntaje: +0

Guardar Cancelar

Feedback

- 1. Webcursos - 9. eCommerce - eBusiness
Link <https://webcursostest.uai.cl/mod/resource/view.php?id=832054>
- 2. Webcursos - 8. CMS - ECM - SCM - BPM

Keywords

- supply chain management
- customer relationship management
- e-commerce
- enterprise content management
- business analytics
- enterprise resource planning

Webcursos	OCW MIT	Merlot
E-Commerce and International Business		https://www.merlot.org/merlot/viewMaterial.htm?id=1350393
E-Commerce Business Ethics Case Study		https://www.merlot.org/merlot/viewMaterial.htm?id=85696
E-Kin2 Case Study		https://www.merlot.org/merlot/viewMaterial.htm?id=75984
Managing the Digital Enterprise		https://www.merlot.org/merlot/viewMaterial.htm?id=90049
Bezbednost podataka u vpn-u na javnoj globalnoj mreži		https://www.merlot.org/merlot/viewMaterial.htm?id=583395
Center for Internet Technologies		https://www.merlot.org/merlot/viewMaterial.htm?id=323803

FIGURA 4. WIDGET CREADO PARA LA REALIZACIÓN DEL EXPERIMENTO EN E-MARKING O TAMBIÉN LLAMADO.

2.4.3 Nuevas características para el estudiante

Una vez las pruebas han sido publicadas por el corrector o profesor, el estudiante al revisar su prueba encontrará por cada pregunta un widget que despliega información sobre el puntaje obtenido y retroalimentación compuesta por un comentario y link a material educativo. En la **Figura 5**, se aprecia la vista que tiene el alumno de su prueba.

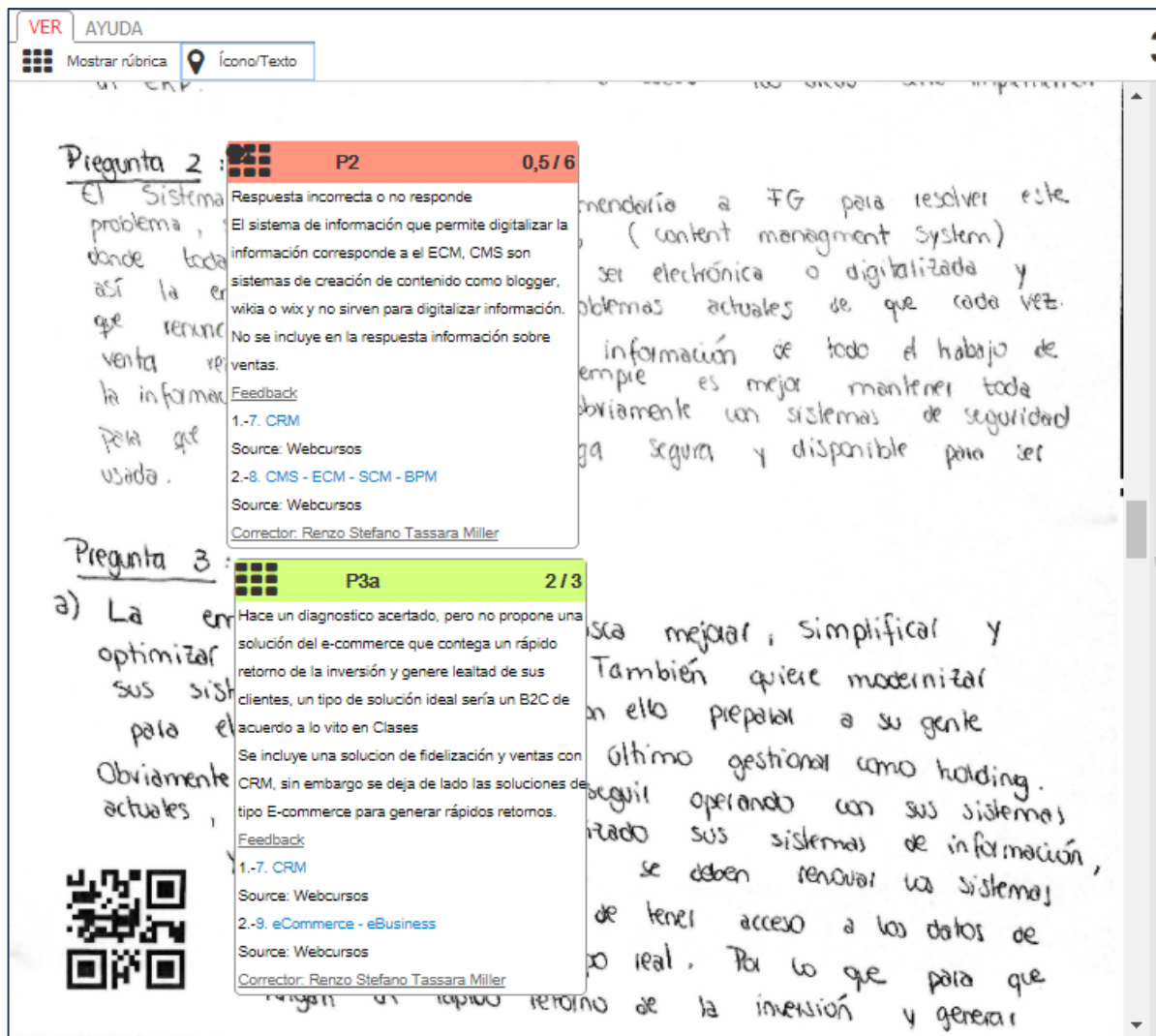


FIGURA 5. PRUEBA DESDE LA VISTA DEL ALUMNO, REVISIÓN DE RETROALIMENTACIÓN.

2.5 Generación de los datos

Para el diseño experimental, que será descrito en el siguiente capítulo, considera la generación de datos cuantitativos y cualitativos. Todos los datos fueron registrados a partir del uso de Emarking y Webcursos, y almacenados en la base de datos correspondiente de Webcursos, la cual, es una base de datos relacional, en este caso, MariaDB. Desde dicha base de datos se realizarán consultas, para posteriormente obtener los datos.

2.6 Métodos para adquisición de datos

Para obtener los datos necesarios, se crearon dos scripts PHP (**Anexo D**), los cuales llevan a cabo consultas SQL a la base de datos MariaDB.

2.7 Métodos para el análisis de datos

En primera instancia, se utilizó el programa MS Excel para consolidar, ordenar los datos y realizar un análisis descriptivo. Posteriormente, se utilizó el programa SPSS Statistics IBM para la normalización de variables, análisis de los supuestos del modelo MANOVA, los cuales consisten en normalidad de las variables, homogeneidad de varianza a nivel de vector y homogeneidad de varianza a nivel de variable, análisis de medias con prueba t-student y, por último, para llevar a cabo los modelos de análisis de la varianza multivariante (MANOVA).

Capítulo 3: Diseño experimental

En este capítulo se define el diseño experimental, incluyendo la pregunta a responder, la hipótesis, selección de variables, restricciones y la selección del modelo experimental.

3.1 Pregunta a responder

Desde el contexto definido en el **Capítulo 1**, la oportunidad y definición de retroalimentación formativa descrito en la **sección 2.1**, nace la pregunta: ¿cuál será el impacto en el aprendizaje del estudiante, al disponer de retroalimentación creada por el corrector usando el nuevo widget enlazado con Open Educational Resources?

3.2 Hipótesis

El objetivo de la investigación consiste en medir y concluir sobre el impacto en el aprendizaje del estudiante cuando la creación de la retroalimentación usa Open Educational Resources.

Desde la pregunta definida en la **sección 3.1** y la investigación de la literatura en la Introducción, se define la hipótesis de la investigación: *El uso de Open Educational Resources por parte del corrector para la creación de retroalimentación y el consumo por parte del estudiante mejora su aprendizaje.*

Esta hipótesis será analizada para cada curso en particular, es decir, tanto para el curso de programación como para el curso de sistemas de información, debido a que per se

asumimos que los estudiantes son diferentes entre ellos, los de primer año de educación superior rondan los 18 años de edad con un alto porcentaje de deserción universitaria, en cambio, los alumnos de último año de universidad, correspondientes al curso de sistemas de información, rondan los 23 años de edad y prácticamente tienen deserción cero, dado que están en la etapa de finalización de los estudios.

3.3 Selección de variables

Para analizar el aprendizaje del alumno se utilizaron las notas de todas las evaluaciones en las cuales se utilizó Emarking y la nota final del curso, que en ambos casos tanto para programación y sistemas de información, la nota final no es explicada completamente por las notas parciales debido a que existen notas por trabajo semestral y notas de menor peso por tareas y/o actividades.

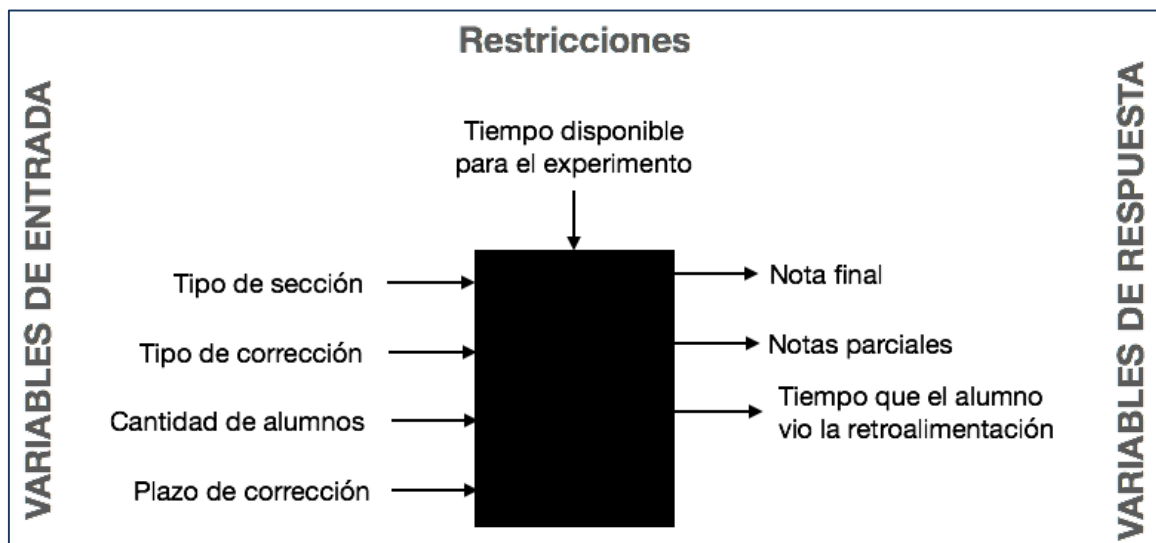


FIGURA 6. DIAGRAMA DE CAJA NEGRA.

En la **Figura 6**, se muestra un diagrama de caja negra para lograr entender de mejor manera las variables de entrada, variables de salida y restricciones del experimento. En variables de entradas se encuentran variables de decisión, que permiten alimentar el experimento, y el resto de variables de entrada corresponden a parámetros o variables

constantes. Por otro lado, las variables de salida corresponden a la respuesta del experimento y las restricciones son el marco general que rige el experimento, en este caso, las únicas restricciones son los plazos o tiempo para realizar acciones.

3.3.1 Variables de entrada

Para controlar el experimento y no aumentar su complejidad innecesariamente, las siguientes variables de entrada se definen como parámetros:

- Cantidad de alumnos: Como se mencionó en sección 1.5, se utilizaron dos cursos, programación y sistemas de información, con cuatro secciones y tres secciones correspondientemente. De este modo, la cantidad de alumnos que cada sección provee es un número fijo, pero no igual para todas. La cantidad fijada de alumnos por sección escapa del control y diseño del experimento, es del ámbito de coordinación de la universidad.
- Plazo de corrección: Definido desde Departamento de Docencia, el plazo para la entrega de la nota y la evaluación corregida es de catorce días a partir desde que se rindió la prueba. Dicho plazo, es igual si la evaluación es corregida por Emarking o a mano sobre el papel original de la prueba.

Las variables de decisión, tipo de sección y tipo de corrección, están diseñadas para analizar los efectos sobre las variables de salida definidas. La **Tabla 1** muestra la configuración respecto al experimento, tanto para programación y sistemas de la información, con respecto a las variables de decisión.

TABLA 1. COMBINACIONES TIPO DE SECCIÓN Y TIPO DE CORRECCIÓN

Sección	Corrección
Programación sección 1 Semestre 1 2017	Emarking normal
Programación sección 2 Semestre 1 2017	Emarking experimental

Programación sección 5 Semestre 1 2017	Emarking experimental
Programación sección 7 Semestre 1 2017	Emarking normal
Sistemas de información sección 2 Semestre 1 2017	Emarking normal
Sistemas de información sección 3 Semestre 1 2017	Emarking normal
Sistemas de información sección 4 Semestre 1 2017	Emarking experimental

3.3.2 Variables de salida

Como se muestra en la **Figura 4**, las variables de salida corresponden a las calificaciones obtenidas por los estudiantes en las pruebas parciales, en programación dentro del set de notas de pruebas parciales se incluye la nota del examen final del curso, en cambio, en sistemas de información no existe un examen dado que existe un proyecto semestral que es tomado como examen. Las notas parciales son obtenidas desde Emarking, una vez que el corrector publica las notas. También, el tiempo que pasa cada alumno revisando las evaluaciones es registrado por Emarking.

Por último, la variable nota final del curso, como se mencionó en la sección 3.3, no es explicada directamente por las notas parciales, debido principalmente por proyectos semestrales que ponderan hacia la nota final de cursos y por tareas y controles llevados a cabo clase a clase, de este modo, la nota final no es combinación lineal de las notas parciales.

3.4 Restricciones y factibilidad del experimento

La principal restricción para realizar el experimento tiene que ver con el tiempo, es un experimento que debe iniciar cuando inicia un semestre académico y finaliza cuando el profesor entrega la nota final del curso. Por lo cual, el iniciar a tiempo es de vital importancia para la correcta recolección de los datos desde la primera evaluación hasta la última.

Por otro lado, una restricción fue el apoyo de la universidad para la realización del experimento, dado que, se realizaría sobre cursos reales que se están impartiendo. En este caso, se contó con el apoyo de la universidad, los profesores y ayudantes para llevar a cabo con éxito el experimento en todas las secciones.

Además, restricciones de tipo presupuestarias no fueron consideradas debido a la universidad se hace cargo del pago hacia los ayudantes correctores de todas las secciones. También, el desarrollo de software que significó la creación del widget con conexión hacia OER de OpenCourseWare de MIT y Merlot.org, así como las nuevas funcionales para profesor, corrector y alumno, **sección 2.4**, fueron desarrolladas en su totalidad, en el contexto de la presente tesis por Hans Cristian Jeria Díaz.

3.5 Selección del modelo experimental

Como se explicó en la **sección 2.7**, el análisis de datos en una primera instancia será exploratorio descriptivo, del cual se desprenderán visualizaciones mediante gráficas y figuras. Para analizar la significancia de lo observado en los datos, se utilizará un modelo de análisis de varianza multivariante (MANOVA), prueba t-student para test de medias, normalización de variables y las correspondientes pruebas de análisis de normalidad y homogeneidad de varianza para cumplir con los supuestos que exige el modelo MANOVA.

Capítulo 4: Metodología

En este capítulo se presenta la metodología definida para el diseño, realización y análisis del experimento, para responder a la pregunta definida en la **sección 3.1** y concluir sobre la hipótesis planteada en la **sección 3.2**. Se describe el diseño y desarrollo de las funcionalidades adicionales incorporadas en Emarking, el entrenamiento de los correctores, la coordinación del experimento en cada sección y por último la adquisición de los datos. El análisis de datos se describe en detalle en el **Capítulo 5**.

4.1 Funcionalidades nuevas para Emarking

En la **sección 2.4**, se explicó la herramienta Emarking y los nuevos desarrollos que se realizaron para lograr las interacciones que se necesitaban en materia de creación y consumo de retroalimentación. De este modo, se decidió desarrollar todo el software requerido al interior del proyecto Emarking y a la vez realizar un aporte al proyecto, que hasta hoy está disponible en él y es de libre acceso para el que lo desee utilizar. A continuación, se detalla el diseño de interfaces y el desarrollo del software.

4.1.1 Diseño

Como se mencionó previamente, el proyecto Emarking es de código abierto⁴ y se decidió sobre él construir los desarrollos necesarios para el experimento. Para almacenar los datos se utilizará la base de datos que soporta a la plataforma

⁴ Repositorio Proyecto Emarking: <https://github.com/villalon/emarking>

Webcursos, los cambios realizados en este nivel fueron la creación de nuevas tablas y nuevos campos en tablas.

La **Figura 7** muestra la interacción a un nivel macro de lo realizado a través de cada evaluación durante el semestre.

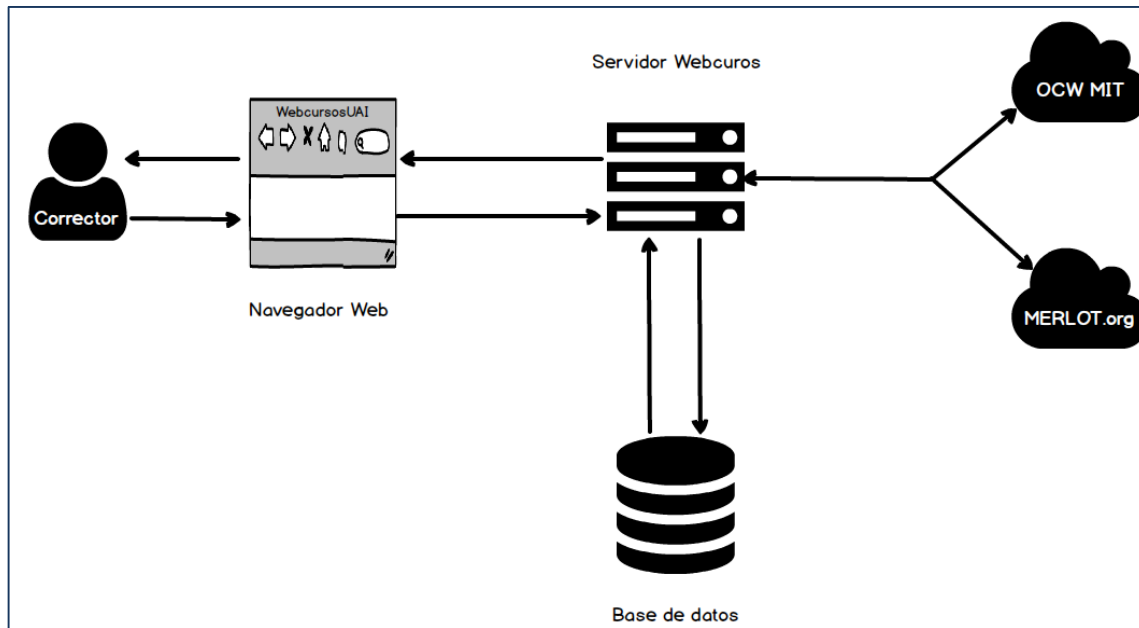


FIGURA 7. DIAGRAMA DE INTERACCIÓN CORRECTOR – OER.

El material desde OpenCourseWare y MERLOT fue obtenido mediante el uso de la librería *PHP Simple HTML DOM Parser*⁵, la cual fue utilizada para realizar Web Scraping⁶ en las páginas de los repositorios. Esto debido a que los repositorios no poseen una *Interfaz de programación de aplicaciones* (Application Programming Interface, API)⁷, lo que imposibilita manipular desde otro software material o búsquedas desde sus páginas web.

⁵ PHP Simple HTML DOM Parser Source: <http://simplehtmldom.sourceforge.net>

⁶ Web Scraping es una técnica utilizada para extraer información de sitios web.

⁷ Una API es un conjunto de funciones y herramientas que pueden ser utilizadas por otro software. <https://www.w3schools.com/>

4.1.2 Desarrollo

Todos los cambios mencionados, que fueron realizados en base al estado actual de Emarking, fueron diseñados y desarrollados por el autor de este documento. Para su implementación se programó en el lenguaje JAVA para el proyecto EmarkingWeb⁸, sub proyecto de Emarking que se encarga de la interfaz de corrección, en lenguaje PHP para el proyecto Emarking y por último en SQL para la creación de nuevas tablas y campos en la base de datos, así como también para la creación de las consultas necesarias para la conexión a la base de datos.

El detalle de los cambios realizados al código fuente puede ser revisado en el registro de cambios que genera la plataforma GitHub que se denominan commit, en las siguientes rutas:

- Para EmarkingWeb <https://github.com/hansnok/eMarkingWeb/commits/master>
- Para Emarking <https://github.com/hansnok/emarking/commits/master>

Cabe destacar que, la programación inició en el mes de octubre del año 2016 y culminó en abril del año 2017.

4.2 Entrenamiento de correctores, grupos control y experimental

Para el desarrollo del experimento no fue necesario reclutar correctores debido a que cada uno de los cursos y secciones seleccionadas posee ayudantes contratados por la universidad, de este modo, los costos asociados a sus actividades son financiados en integra medida por la universidad. Además, cada ayudante corrector aceptó voluntariamente participar en el experimento, aceptando que en algunos casos aquello

⁸ Fuente EmarkingWeb: <https://github.com/villalon/eMarkingWeb>

significaría una mayor carga de trabajo, principalmente en la creación de retroalimentación.

Previo a la realización de la prueba 1, tanto en programación como en sistemas de información, se llevó a cabo una charla en conjunto con los correctores de grupo control y experimental. Con respecto a aquello, el fin de llevar a cabo una charla conjunta es reducir el sesgo respecto a la información entregada, es decir, que la única diferencia sea la herramienta que usarán ambos grupos y no el entrenamiento diferenciado.

En **Anexo F**, se puede encontrar la presentación que se realizó a los correctores, el enfoque fue aplicado de forma práctica y explicando que es un marco de acciones y que se debe tomar como buenas prácticas en vez de una guía rígida de lo que se debe hacer.

4.3 Coordinación de experimento

Debido a que se utilizaron dos cursos y siete secciones, a continuación, en la **Tabla 2** se detallan las fechas en las cuales ocurrieron todas las evaluaciones, tanto para el curso programación como para el curso de sistemas de información.

TABLA 2. FECHAS DE EVALUACIONES PROGRAMACIÓN Y SISTEMAS DE INFORMACIÓN.

Sección	Fecha Evaluaciones
Programación Prueba 1	22/04/2017
Programación Prueba 2	27/05/2017
Programación Examen	06/06/2017
Sistemas de información Prueba 1	24/04/2017
Sistemas de información Prueba 2	16/05/2017
Sistemas de información Prueba 3	20/06/2017

4.4 Adquisición de datos

En la **sección 2.6**, se mencionó la creación de dos scripts que se ejecutaron en el servidor de la plataforma Webcursos para adquisición de datos. Una vez terminada todas las evaluaciones, aproximadamente entre septiembre – octubre 2017 se adquirieron los datos generados a través de todas las evaluaciones del semestre, así como la nota final obtenida por cada estudiante.

Los datos fueron obtenidos en archivos CSV (*Comma Separated Values*), a través de los scripts mencionados en **sección 2.6**, para su posterior limpieza y análisis. El siguiente capítulo describirá la fase final del experimento que corresponde al análisis de los resultados.

Capítulo 5: Análisis de resultados

En este capítulo se presenta el análisis a partir de los datos obtenidos luego de la realización del experimento. El objetivo es concluir sobre el uso de la herramienta y su impacto en el aprendizaje de los estudiantes. Se realizará un análisis por cada curso, es decir, se realizarán conclusiones para cada uno de ellos, porque el comportamiento de los alumnos es diferente para cada curso, principalmente por la diferencia de madurez entre un alumno de primer año versus un alumno de último año.

Para el curso de Programación, el total de alumnos utilizados para el experimento, contando control y experimental, fue de 176. Para el curso de Sistemas de Información, el total de alumnos fue de 93.

5.1 Acontecimientos relevantes

Antes de analizar los resultados, resulta contingente describir el proceso llevado a cabo durante el primer semestre del año 2016, periodo en el cual se realizó el experimento, debido a que en cierto grado hubo acontecimientos que afectaron el proceso.

En primer lugar, el ramo de programación al ser de primer semestre de educación superior, posee un alto porcentaje de reprobación, rondando entre el 30% a 60%. De este modo, durante el semestre se tomaron decisiones a nivel de coordinación que afectaron el experimento. Las principales situaciones fueron:

- En un inicio la corrección de las pruebas sería mediante modalidad cruzada, es decir, los ayudantes correctores no corregirían la misma sección a la cual pertenecían, sino corregirían una sección al azar y de manera anónima. Esta modalidad solo fue utilizada en la primera prueba y luego desechada para las posteriores.

- En consecuencia, a lo anterior se suma el contexto descrito previamente de un alto grado de reprobación, por lo cual, las pruebas corregidas, por orden de coordinación, debían ser entregadas a lo máximo tres días de ser rendida por los estudiantes. De este modo, en palabras de los correctores experimentales: *la herramienta tiene un potencial que no se pudo aprovechar en su totalidad debido al escaso tiempo para corregir las pruebas.*

El condicionante principal para el curso de programación fue claramente el nivel de reprobación que este presenta, que llevo a coordinación a tomar las medidas que se describieron previamente.

Por otro lado, el curso de sistemas de información se llevó a cabo según lo planificado desde el inicio. Las pruebas corregidas fueron entregadas luego de dos semanas de ser rendidas por los estudiantes y además siempre fueron corregidas por los correctores designados para cada sección. En este caso, no hubo variación desde lo planificado y los comentarios de los ayudantes fueron que la herramienta era fácil e intuitiva de usar y, que dado el tiempo para corregir no era demasiada carga extra.

5.2 Análisis del curso de programación

A continuación, se describe todo el proceso de análisis llevado a cabo para el curso de programación. Como se mencionó en la **sección 2.7**, se seleccionó el modelo MANOVA para el análisis. Este modelo es una generalización de modelos de comparación de medias y permite, si se aprecian diferencias, concluir en que sí existe relación entre las variables relacionadas. En particular con el análisis MANOVA, se analizan diferencias entre grupos a partir de múltiples variables dependientes métricas.

Para contrastar los supuestos del modelo MANOVA, es necesario llevar a cabo tres pruebas estadísticas:

- Prueba Kolmogorov-Smirnov, se debe realizar para cada una de las variables a ser utilizadas por el modelo MANOVA y permite certificar que la distribución de las variables son distribución normal. De ser rechazada la hipótesis, se debe aplicar la transformación correspondiente a la variable o de lo contrario no es posible su utilización en el modelo.
- Prueba de Levene, la cual permite certificar que la varianza en cada una de las subpoblaciones o grupos es la misma o, dicho de otro modo, que las varianzas son heterogéneas.
- Prueba de Box, la cual permite certificar que las matrices de covarianza generadas para cada nivel de tratamiento son idénticas a nivel poblacional. Este supuesto se puede considerar como la generalización multivariante del supuesto de igualdad de varianzas de error, es decir, la generalización de la prueba de Levene.

Para el correcto cumplimiento de la prueba de Kolmogorov-Smirnov, fue necesaria la normalización de múltiples variables, ya sea, para el modelo MANOVA del curso de Programación, como también para el curso de Sistemas de Información. Para llevar a cabo dicha labor, se utilizó un método denominado *A Two-Step Approach for Transforming Continuous Variables to Normal* desarrollado por Gary Templeton [18], el cual consiste en dos transformaciones realizadas a una variable continua, primero de rango fraccional y luego la aplicación de la función inversa a la distribución normal. Los resultados fueron satisfactorios debido a que en todos los casos donde se utilizó el método se cumplió con el test de normalidad.

En las siguientes secciones, cada una de las pruebas mencionadas serán realizadas de manera explícita, mencionando los correspondientes test de hipótesis subyacentes.

5.2.1 Variables, normalización y supuestos del modelo MANOVA

Las variables dependientes seleccionadas para el análisis de este curso fueron:

- Nota prueba 1.
- Tiempo de revisión, por parte del alumno, en prueba 1.
- Nota prueba 2.
- Tiempo de revisión, por parte del alumno, en prueba 2.
- Nota final del curso.
- Diferencia entre nota de prueba 2 y prueba 1.
- Diferencia entre nota fina y prueba 2.

La variable independiente seleccionada para el análisis de este curso fue:

- Tipo, experimental o control.

Cabe destacar que, las variables correspondientes a tiempo de revisión de examen y nota del examen resultaron no ser estadísticamente relevantes para el modelo, y esto es explicado porque, siendo el examen la evaluación final sin replica de más notas, su influencia en el aprendizaje del alumno es muy baja y, la retroalimentación de este no aporta en comparación a las demás evaluaciones.

La **Tabla 3**, muestra los estadísticos descriptivos de las variables presentadas previamente, la información entregada corresponde es la media y desviación estándar. El total de alumnos control fue de 85 y de alumnos experimentales fue de 91.

TABLA 3. ESTADÍSTICOS DESCRIPTIVOS. TIEMPO MEDIDO EN SEGUNDOS.

Variable	Tipo	Media	Desviación estándar
Nota prueba 1	Control	3,90	0,95
	Experimental	3,68	0,93

Tiempo revisión prueba 1	Control	1.689,24	10.812,96
	Experimental	8.597,01	9.835,47
Nota prueba 2	Control	4,30	1,41
	Experimental	4,12	1,30
Tiempo revisión prueba 2	Control	5.381,89	12.970,10
	Experimental	8.450,82	14.849,13
Nota final	Control	4,85	1,22
	Experimental	4,44	1,38
Diferencia prueba 2 y prueba 1	Control	0,44	0,87
	Experimental	0,64	0,95
Diferencia nota final y prueba 2	Control	0,58	0,65
	Experimental	0,33	0,82

Como se mencionó en la **sección 5.2**, se normalizaron todas las variables continuas. Para efecto de este análisis fue aplicado el método a todas las variables dependientes. En la **Tabla 4**, se aprecian los resultados correspondientes a la prueba Kolmogorov-Smirnov, en todos los casos no es posible rechazar la hipótesis nula, $p\text{-valor} > 0,05$, que hace mención a que las variables provienen de una distribución normal.

TABLA 4. PRUEBA DE NORMALIDAD KOLMOGOROV-SMIRNOV.

Tipo	Estadístico de prueba	Sig. Asintótica(bilateral)
Nota prueba 1	0,036	0,20
Tiempo revisión prueba 1	0,049	0,20
Nota prueba 2	0,025	0,20
Tiempo revisión prueba 2	0,023	0,20
Nota final	0,042	0,20
Diferencia prueba 2 y prueba 1	0,037	0,20
Diferencia nota final y prueba 2	0,040	0,20

La prueba Box, que se observa en la **Tabla 5**, proporciona un valor de 0,088 el cual es mayor al nivel de significancia 0.05, por lo cual, no es posible rechazar la hipótesis nula. De este modo, las covarianzas son homogéneas a nivel poblacional.

TABLA 5. PRUEBA BOX.

M de Box	40,29
Estadístico de prueba	1,378
Significancia	0,088

Por último, la prueba de Levene, que se observa en la **Tabla 6**, contrasta la hipótesis de no heterogeneidad de varianzas por grupos. En este caso, para todas las variables el valor obtenido por la prueba es mayor al nivel de significancia 0.05, por lo cual, no es posible rechazar la hipótesis nula, y se acepta la hipótesis de igualdad de varianza por grupos.

TABLA 6. PRUEBA LEVENE.

Variable dependiente	Estadístico de Levene	Significancia
Nota prueba 1	0,072	0,789
Tiempo revisión prueba 1	0,675	0,412
Nota prueba 2	0,661	0,417
Tiempo revisión prueba 2	3,511	0,063
Nota final	1,347	0,247
Diferencia prueba 2 y prueba 1	1,259	0,263
Diferencia nota final y prueba 2	4,622	0,051

Una vez que los supuestos necesarios para la realización del análisis MANOVA han sido comprobados estadísticamente, es posible proseguir con el estudio. En la siguiente sección se describe el modelo MANOVA, combinación de variables y se analizarán los resultados.

5.2.2 Modelo MANOVA y análisis de resultados

El modelo MANOVA utilizado para el curso de programación, tiene interés en explicar el tipo de evaluación, sección experimental o control, mediante las variables dependientes. El modelo se presenta en la **Figura 8**.

Tipo ~ *Nota Prueba 1, Tiempo Revisión Prueba 1, Nota Prueba 2, Tiempo Revisión Prueba 2, Nota Final, Diferencia entre Prueba 2 y Prueba 1, Diferencia entre Nota Final y Prueba 2*

FIGURA 8. MODELO MANOVA.

En la **Tabla 7**, se aprecian las pruebas multivariantes a las cuales fue sometido el modelo presentado previamente. En los cuatro casos, Traza de Pillai, Lamba de Wilks, Traza de Hotelling y Raíz mayor de Roy el modelo es aceptado, dado que, el valor de significancia del modelo es menor al valor 0.05. De este modo, es posible realizar el análisis multivariado para analizar como variable independiente el tipo evaluación.

TABLA 7. MODELO MULTIVARIANTE.

Modelo	Valor	F	Significancia
Traza de Pillai	0,187	5,514	0.0
Lamba de Wilks	0,813	5,514	0.0
Traza de Hotelling	0,230	5,514	0.0
Raíz mayor de Roy	0,230	5,514	0.0

Un análisis posterior, es el análisis de pruebas de efectos inter-sujetos, que permite descubrir las variables estadísticamente más influyentes al modelo. En la **Tabla 8**, se aprecia que las variables: *Tiempo Revisión Prueba 1, Nota Prueba 1, Nota final y Diferencia entre Nota Final y Prueba 2*, son las variables más influyentes al modelo.

TABLA 8. PRUEBAS DE EFECTO INTER-SUJETO ANOVA.

Variable dependiente	F	Significancia
Nota prueba 1	8,79	0,003

Tiempo revisión prueba 1	19,70	0,000
Nota prueba 2	0,73	0,394
Tiempo revisión prueba 2	2,12	0,147
Nota final	4,29	0,040
Diferencia prueba 2 y prueba 1	2,22	0,139
Diferencia nota final y prueba 2	4,83	0,029

A continuación, se presentan los histogramas de las cuatro variables más influyentes en el modelo para observar su comportamiento.

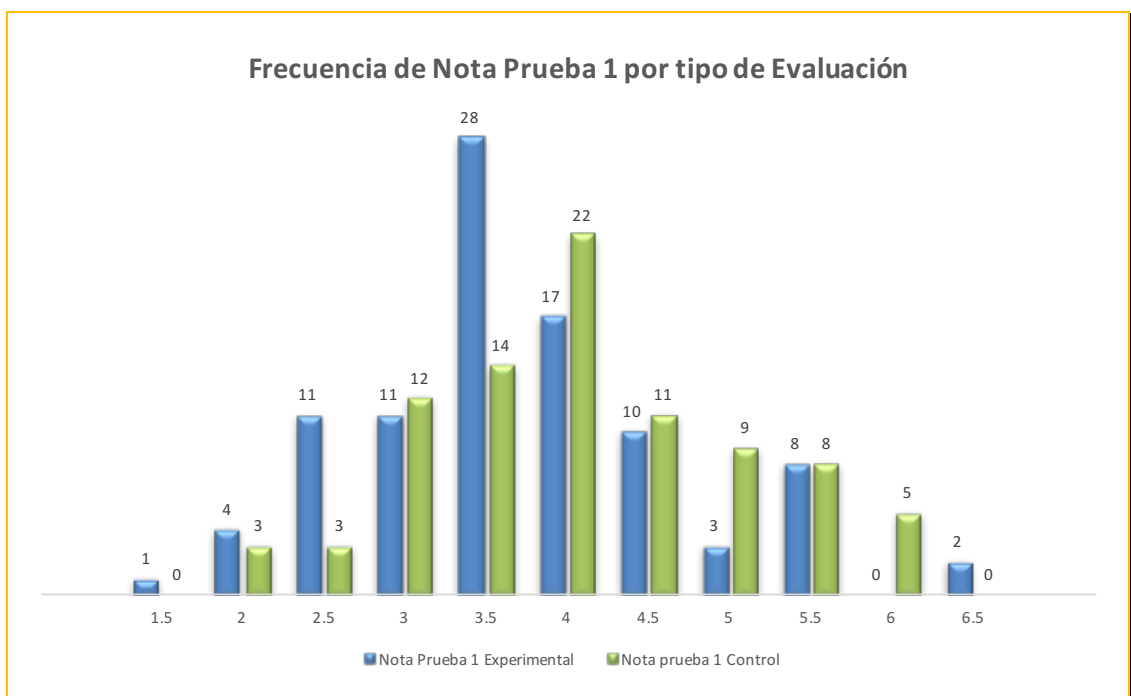


GRÁFICO 1. HISTOGRAMA DE FRECUENCIAS PARA NOTA DE PRUEBA 1 POR TIPO DE EVALUACIÓN.

Es posible observar que existe una mayor cantidad de notas bajo 4 en los cursos experimentales. También, se observa que ambas distribuyen normal pero la media de los cursos experimentales está cercana a 3.5, mientras que la media de los grupos control está cercano a 4. Para poder concluir respecto a esta observación, es necesario realizar un test de medias, en este caso T-Student. En la **Tabla 9**, se aprecia el resultado obtenido de la prueba.

TABLA 9. PRUEBAS T-STUDENT PARA NOTA PRUEBA 1.

	Prueba de Levene		Prueba t			
	F	Sig.	t	Sig.	Inferior	Superior
Se asumen varianzas iguales	0,139	0,710	2,862	0,005	0,128	0,698
No se asumen varianzas iguales			2,863	0,005	0,128	0,698

La prueba, en primera instancia discrimina si las muestras que se están comparando poseen o no iguales varianzas, por lo cual, lleva a cabo una prueba de Levene, la cual resultó verdadera con un nivel de 0,710, de este modo se prosigue por la fila “Se asumen varianzas iguales”. Posteriormente, se evalúa el nivel de significancia de la prueba t, en este caso p-valor > 0,005, por lo cual indica que existen diferencias significativas entre las medias de ambas muestras.

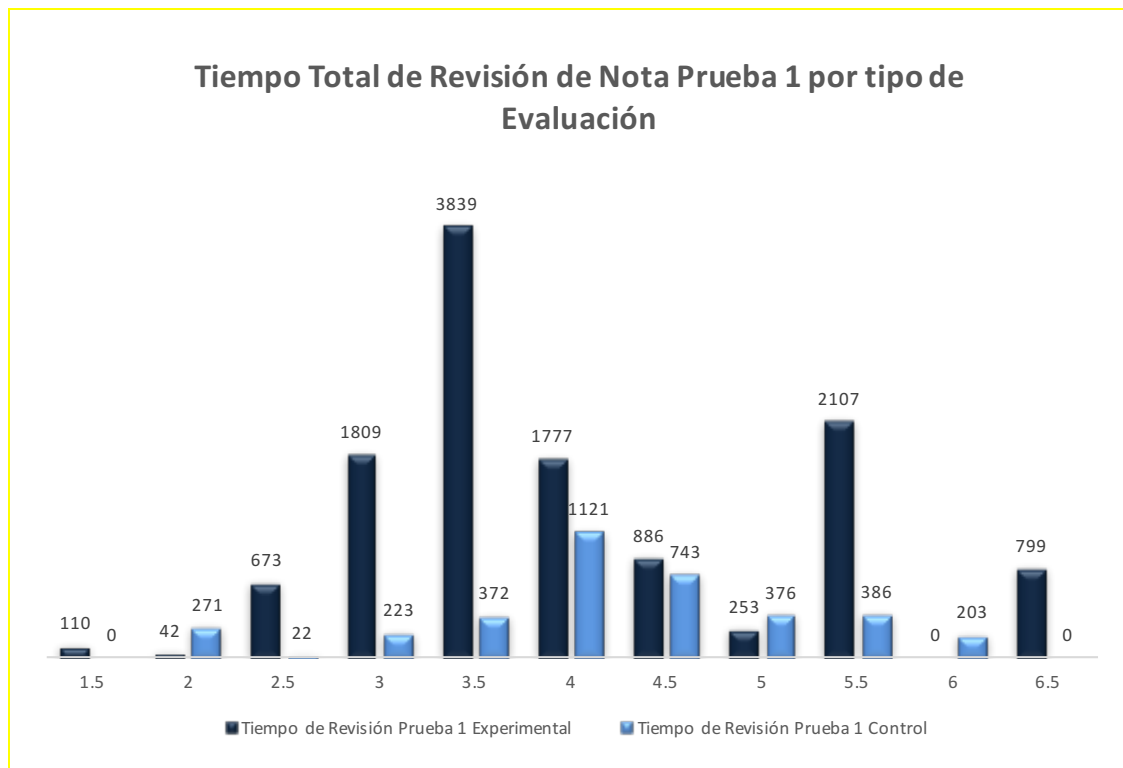


GRÁFICO 2. HISTOGRAMA DE SUMA TOTAL DE TIEMPO DE REVISIÓN PRUEBA 1 POR RANGO DE NOTA OBTENIDA Y POR TIPO DE EVALUACIÓN. TIEMPO MEDIDO EN MINUTOS.

El **Gráfico 2** está relacionado con el **Gráfico 1** debido a que hace referencia a las mismas pruebas en su respectivo nivel, por ejemplo, las 28 pruebas de los alumnos que se encuentran en el rango 3,5 de curso experimental de **Gráfico 1** estuvieron 3839 minutos revisando sus pruebas.

En este caso, no es necesario realizar un test de medias, principalmente porque se observa a simple vista que los cursos experimentales superan ampliamente a los cursos control en tiempo de revisión de sus pruebas. Cabe destacar que en la prueba 1 todos los cursos usan la herramienta Emarking normal, aun no se había influido el widget con conexión a OER. Por lo cual, es un dato relevante que en las mismas condiciones existan diferencias tan grandes en tiempos de revisión, más aún cuando es primera vez en su vida que los estudiantes tenían contacto con Emarking.

Como dato extra, el total de minutos de revisión para los cursos experimentales fue de 12.295 minutos versus 3.717 minutos para la revisión de sus pruebas 1 de programación.

En el **Gráfico 3**, se aprecia un pick en los cursos experimentales en el rango de nota 4–4,4 versus el pick que se observó en el **Gráfico 1**, donde ocurría en rango de nota 3,1–3,5, es decir, existió una translación de rango y por ende un aumento de rendimiento. Además, se observa una distribución más uniforme luego del rango 4 que en la prueba 1 no existía. En este caso, también es atinente realizar un test de medias con el fin de esclarecer si existe diferencia estadísticamente significativa entre las medias de ambos tipos de cursos.

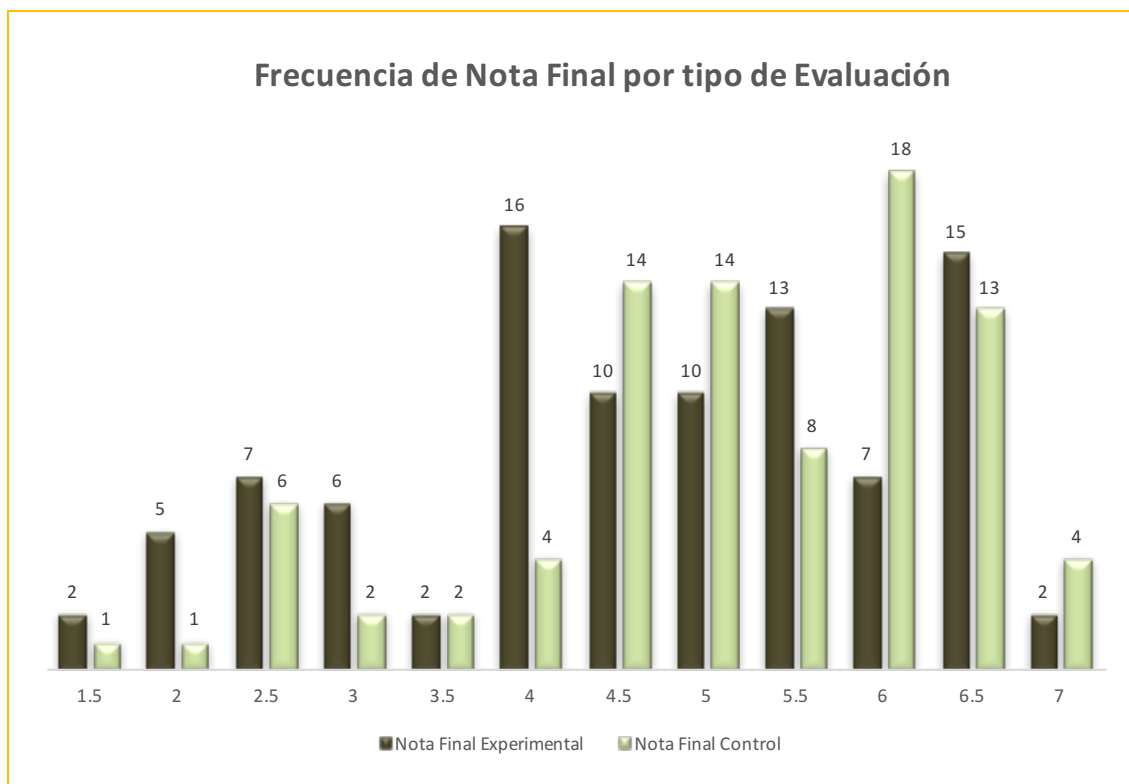


GRÁFICO 3. HISTOGRAMA DE FRECUENCIA DE NOTA FINAL POR TIPO DE EVALUACIÓN.

TABLA 10. PRUEBAS T-STUDENT PARA NOTA FINAL.

Prueba de Levene				Prueba t			
	F	Sig.		t	Sig.	Inferior	Superior
Se asumen varianzas iguales	2,295	0,132		2,126	0,035	0,308	0,825
No se asumen varianzas iguales				2,139	0,034	0,332	0,822

La prueba de Levene asume varianzas iguales, por lo cual, se prosigue con la fila correspondiente. La prueba t resulta con p-valor > 0,034 que indica que existen diferencias significativas entre las medias de ambas muestras.

Desde prueba 1 hasta la nota final, observando la **Tabla 10** y **Tabla 11**, es posible observar una disminución entre las diferencias que existen entre los cursos

experimental y control, no en un grado que no sea posible estadísticamente diferenciarlas, pero hubo un acercamiento, que además es posible explicarlo por el cambio de rango que se apreció en el Gráfico 3, donde la gran concentración de notas de los cursos experimentales pasó de un estado de reprobación a un estado de aprobación.

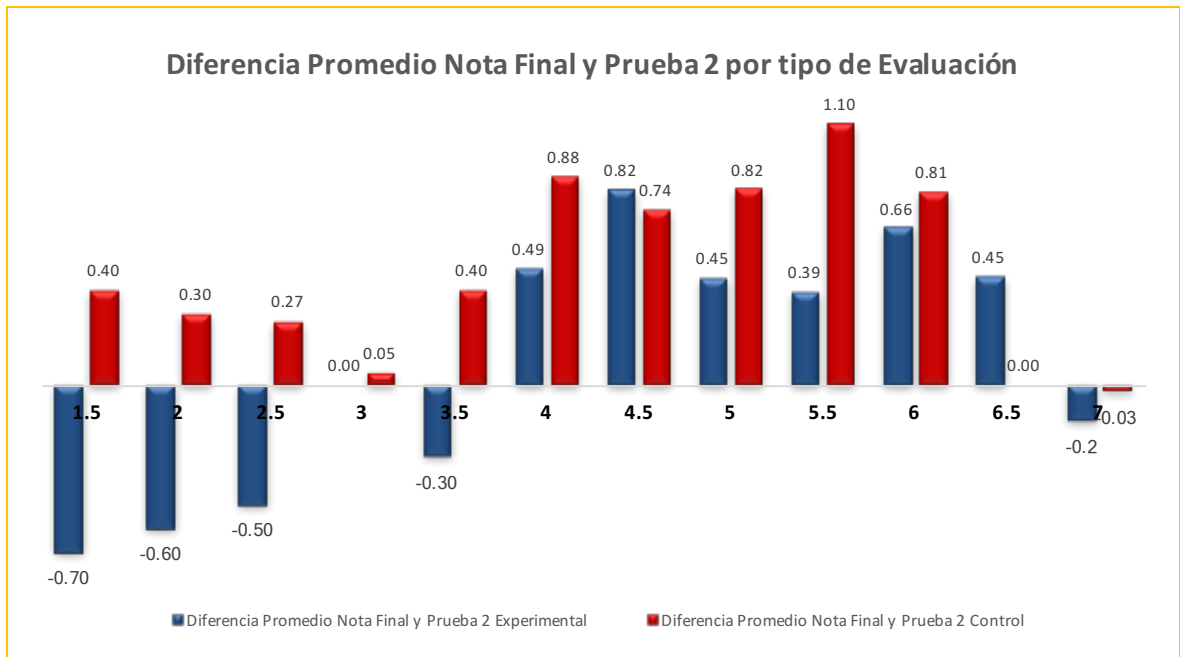


GRÁFICO 4. HISTOGRAMA DIFERENCIAS PROMEDIO ENTRE NOTA FINAL Y PRUEBA 2 RESPECTO A LA NOTA FINAL OBTENIDA POR EL ALUMNO Y POR TIPO DE EVALUACIÓN.

El **Gráfico 4** está relacionado con el **Gráfico 3**, debido a que es construyó a partir del **Gráfico 3** y las notas finales, por lo cual, las posiciones de las diferencias hacen referencia a las notas finales es obtenidas, por ejemplo, el valor 0,81 de diferencia en nivel 6 de curso control en **Gráfico 4**, es el promedio de diferencias que tuvieron las 18 notas finales que se señalan en el **Gráfico 3** en el mismo nivel 6. De este modo, es posible entrar en mayor profundidad en la información.

Se observa claramente que la diferencia siempre fue positiva para los cursos experimentales, excepto por un pequeño valor negativo en el rango 6,5-7,0, en cambio, para el caso de los cursos experimentales las notas decrecieron en los rangos más

bajos, es decir, a los alumnos que en la prueba 1 tuvieron un mal rendimiento no lograron progresar para el final del curso.

5.2.3 Conclusiones

No es posible concluir que el uso del widget con conexión a recursos OER ayudo a los alumnos de los cursos experimentales de programación con conclusiones estadísticamente significativas, esto debido a la información y resultados expuestos en la **sección 5.2.1**.

Este comportamiento es posible explicarlo por los eventos relatados en la **sección 5.1**, donde los tiempos de corrección se vieron disminuidos en un 78%, esto influyendo directamente en la calidad de la retroalimentación generada por los correctores, donde el esfuerzo se centró en realizar la mejor calificación posible en el acotado tiempo que disponían en vez de generar una buena retroalimentación.

Desde el punto de vista de los alumnos, ocurrió una situación similar, donde las pruebas eran rendidas un día sábado, las notas obtenidas el día lunes y debían tomar decisiones de corrección, cambio de proyecto y rendición de prueba recuperativa al próximo día. En esta modalidad, poco importa al alumno revisar su prueba si está pensando en las próximas variables que evaluarán en el curso.

Una segunda mirada a los resultados de forma más global, es posible inferir que el curso de programación no es un curso maduro aun en términos de coordinación, de modo que, implementar mejoras para la realización de experimentos y análisis no es una buena opción. Quizás aplicando un enfoque netamente en el curso de programación, sus debilidades y fortalezas propias es posible obtener un mejor resultado y además una herramienta que se adecue sobre todo a sus necesidades y problemáticas.

5.3 Análisis curso de sistemas de información

De igual modo que la **sección 5.2**, para el análisis del curso de Sistemas de Información, se seleccionó el modelo MANOVA. Por lo tanto, las variables utilizadas para este modelo deben cumplir los supuestos ya descritos, también en **sección 5.2**.

En las siguientes secciones, se describen las variables, los supuestos y sus validaciones, el modelo MANOVA y la conclusión respecto al experimento en el contexto del curso sistemas de información.

5.3.1 Variables, normalización y supuestos del modelo MANOVA

Las variables dependientes seleccionadas para el análisis de este curso fueron:

- Tiempo de revisión, por parte del alumno, en la prueba 1.
- Tiempo de revisión, por parte del alumno, en la prueba 2.
- Tiempo de revisión, por parte del alumno, en la prueba 3.
- Nota final del curso.
- Diferencia entre nota prueba 3 y prueba 1.
- Diferencia entre nota final y prueba 1.

La variable independiente seleccionada para el análisis de este curso fue:

- Tipo, experimental o control.

Cabe destacar que, las variables correspondientes a notas de prueba 1, 2 y 3 no fueron estadísticamente significativas para el modelo, pero si fue estadísticamente significativa la combinación lineal de ellas como, por ejemplo, la diferencia entre la nota final del curso y la nota de la prueba 1. Este comportamiento es posible explicarlo debido a que alumnos de último año tienen un rendimiento parecido, por lo tanto,

entre grupos no existe diferencia estadística significativa entre medias y varianzas respecto a las notas obtenidas.

La **Tabla 11**, muestra los estadísticos descriptivos de las variables presentadas previamente, la información entregada corresponde a la media, desviación estándar y cantidad. El total de alumnos control fue de 39 y de alumnos experimentales de 54.

TABLA 11. ESTADÍSTICOS DESCRIPTIVOS. TIEMPOS MEDIDOS EN SEGUNDOS.

Variable	Tipo	Media	Desviación estándar
Tiempo revisión prueba 1	Control	1.928,36	2.686,91
	Experimental	1.407,04	2.646,33
Tiempo revisión prueba 2	Control	1.164,87	2.116,89
	Experimental	414,91	1.835,61
Tiempo revisión prueba 3	Control	1.243,25	1.974,34
	Experimental	468,50	1.923,80
Nota final	Control	5,167	0,592
	Experimental	5,492	0,709
Diferencia prueba 3 y prueba 1	Control	-0,353	0,916
	Experimental	0,011	0,842
Diferencia nota final y prueba 1	Control	0,071	0,542
	Experimental	0,499	0,603

Igual que en la **sección 5.2** del curso de programación, se normalizaron todas las variables continuas, para este caso, se normalizaron todas las variables dependientes. En la **Tabla 12**, se aprecian los resultados correspondientes a la prueba Kolmogorov-Smirnov, en todos los casos no es posible rechazar la hipótesis nula, $p\text{-value} > 0.05$, que hace mención a que las variables provienen de una distribución normal.

TABLA 12. PRUEBA DE NORMALIDAD KOLMOGOROV-SMIRNOV.

Tipo	Estadístico de prueba	Sig. Asintótica(bilateral)
Tiempo revisión prueba 1	0,026	0,20
Tiempo revisión prueba 2	0,042	0,20
Tiempo revisión prueba 3	0,033	0,20
Nota final	0,046	0,20
Diferencia prueba 3 y prueba 1	0,026	0,20
Diferencia nota final y prueba 1	0,019	0,20

La prueba Box, que se observa en la **Tabla 13**, proporciona un valor de 0,221 el cual es mayor al nivel de significancia 0.05, por lo cual, no es posible rechazar la hipótesis nula. De este modo, las covarianzas son homogéneas a nivel poblacional.

TABLA 13. PRUEBA BOX.

M de Box	22,30
Estadístico de prueba	0,984
Significancia	0,480

Por último, la prueba de Levene, que se observa en la **Tabla 14**, contrasta la hipótesis de no heterogeneidad de varianzas por grupos. En este caso, para todas las variables el valor obtenido por la prueba es mayor al nivel de significancia 0.05, por lo cual, no es posible rechazar la hipótesis nula, y se acepta la hipótesis de igualdad de varianza por grupos.

TABLA 14. PRUEBA LEVENE.

Variable dependiente	Estadístico de Levene	Significancia
Tiempo revisión prueba 1	0,007	0,931
Tiempo revisión prueba 2	1,334	0,251
Tiempo revisión prueba 3	0,056	0,814
Nota final	1,382	0,243
Diferencia prueba 3 y prueba 1	0,723	0,397

Diferencia nota final y prueba 1	0,510	0,477
----------------------------------	-------	-------

Una vez, los supuestos necesarios para la realización de un análisis MANOVA han sido comprobados estadísticamente, es posible proseguir con el estudio. En la siguiente sección se describe el modelo MANOVA, combinación de variables y se analizaran los resultados.

5.3.2 Modelo MANOVA y análisis de resultados

El modelo MANOVA utilizado para el curso de sistemas de información, tiene interés en explicar el tipo de evaluación, sección experimental o control, mediante las variables dependientes. El modelo se presenta en la **Figura 9**.

Tipo ~ *Tiempo Revisión Prueba 1, Tiempo Revisión Prueba 2, Tiempo Revisión Prueba 3, Nota Final, Diferencia entre Prueba 3 y Prueba 1, Diferencia entre Nota Final y Prueba 1*

FIGURA 9. MODELO MANOVA.

En la **Tabla 15**, se aprecian las pruebas multivariante a las cuales fue sometido el modelo presentado previamente. En los cuatro casos, Traza de Pillai, Lambda de Wilks, Traza de Hotelling y Raíz mayor de Roy el modelo es aceptado con un valor, dado que el valor de significancia del modelo es menor al valor 0.05.

TABLA 15. MODELO MULTIVARIANTE.

Modelo	Valor	F	Significancia
Traza de Pillai	0,252	4,817	0.000
Lambda de Wilks	0,748	4,817	0.000
Traza de Hotelling	0,336	4,817	0.000
Raíz mayor de Roy	0,336	4,817	0.000

Un análisis posterior, es el análisis de pruebas de efectos inter-sujetos, que permite descubrir las variables estadísticamente más influyentes al modelo. En la **Tabla 16**, se aprecia que las variables: *Tiempo revisión prueba 3*, *Nota final* y *Diferencia nota final y prueba 1*. Son las variables más influyentes al modelo.

TABLA 16. PRUEBAS DE EFECTO INTER-SUJETO ANOVA.

Variable dependiente	F	Significancia
Tiempo revisión prueba 1	2,105	0,150
Tiempo revisión prueba 2	3,870	0,052
Tiempo revisión prueba 3	5,973	0,016
Nota final	5,449	0,022
Diferencia prueba 3 y prueba 1	3,928	0,051
Diferencia nota final y prueba 1	12,411	0,001

A continuación, se presentan los histogramas de las cuatro variables más influyentes en el modelo para observar su comportamiento respecto al tipo de evaluación y control y experimental.

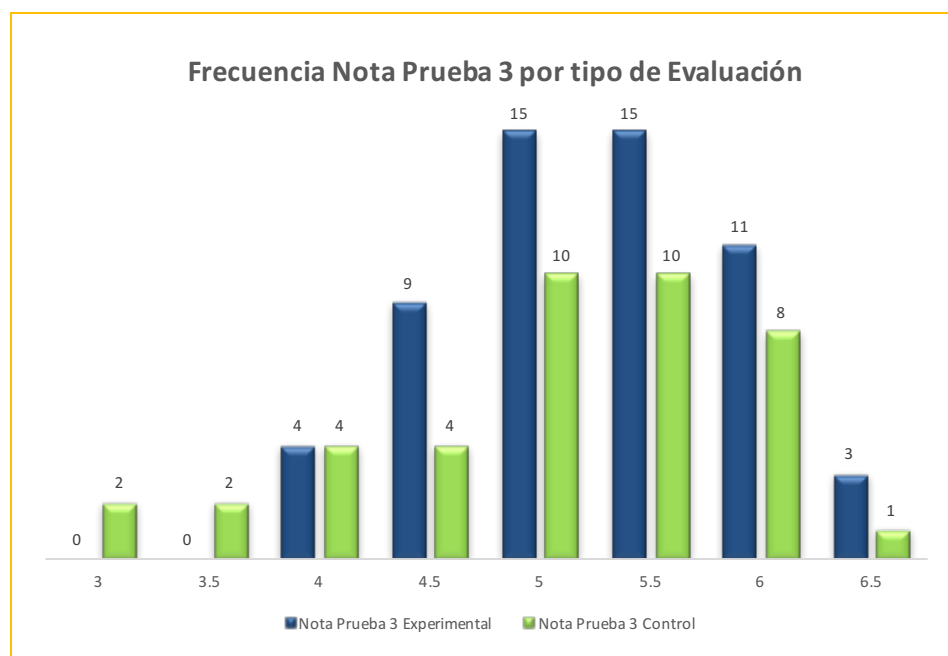


GRÁFICO 5. HISTOGRAMA DE FRECUENCIAS DE NOTA PRUEBA 3 POR TIPO DE EVALUACIÓN.

Para una mejor evaluación de la variable tiempo de revisión prueba 3 se expondrá la gráfica de frecuencias de las notas de la prueba 3.

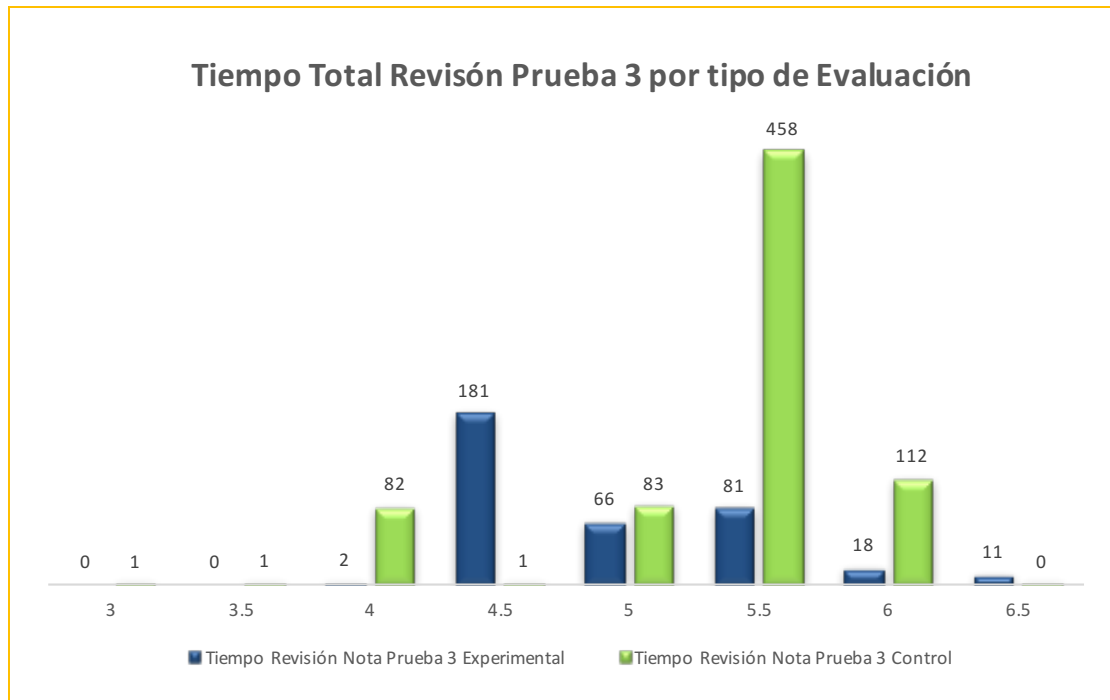


GRÁFICO 6. HISTOGRAMA DE TIEMPO TOTAL DE REVISIÓN DE NOTA PRUEBA 3 POR TIPO DE EVALUACIÓN. TIEMPO MEDIDO EN MINUTOS.

En el **Gráfico 6** es posible observar un alto pick en el rango 5,1-5,5, explorando los datos dicho pick es explicado por el tiempo de revisión de dos alumnos en el curso control que registraron 193 y 173 minutos cada uno. Sin quitar ambos estudiantes del histograma, los tiempos totales de revisión son 359 minutos para experimental y 739 minutos para control, quitando a los mencionados estudiantes, el tiempo de control decae hasta 373 minutos. De este modo, al tener datos atípicos es posible una alta significancia con el modelo MANOVA, debido principalmente porque datos atípicos alteran en gran medida el valor de la varianza de dicha variable.

Por lo cual, es posible observar la misma cantidad de tiempo de revisión para control y experimental en revisión de prueba 3, lo cual es esperable, principalmente por dos hechos. En primer lugar, la prueba 3 (**Anexo C.11**) estaba compuesta de 6 preguntas de

desarrollo con 4 hojas para respuestas, lo que la convirtió en una prueba relativamente corta, tanto para corregir como para ser revisada por el alumno y, en segundo lugar, los alumnos tienen una experiencia de 5 años con la herramienta Emarking, saben cómo navegar, revisar el resumen de la rúbrica e ir directo a las preguntas que tienen dudas sobre la corrección y leer lo que el corrector generó para ellos.

En el **Gráfico 5** si es posible observar una gran diferencia respecto a frecuencia en niveles sobre 4 entre grupo experimental y control, de este modo resulta necesario realizar una prueba de medias para discriminar si existe evidencia estadística que soporte dicha afirmación.

TABLA 17. PRUEBAS T-STUDENT PARA NOTA PRUEBA 3.

	Prueba de Levene		Prueba t			
	F	Sig.	t	Sig.	Inferior	Superior
Se asumen varianzas iguales	1,698	0,196	-0,501	0,617	-0,446	0,266
No se asumen varianzas iguales			-0,488	0,617	-0,457	0,277

El caso presentado en la **Tabla 17** respecto a la prueba de medias, ejemplifica porque el modelo MANOVA no eligió la variable dependiente *Nota prueba 3* como una variable influyente. En primer lugar, se acepta la hipótesis de varianzas iguales y posteriormente $p\text{-valor} < 0,05$, por lo cual, no es posible rechazar la hipótesis de igualdad de medias. De este modo, no existe evidencia estadística para rechazar que la media de los cursos control y experimental es diferente en nota de prueba 3.

El promedio de nota prueba 3 para control fue de 4,7 mientras que para experimental fue de 4,9. Por la evidencia mencionada, tanto la nota de prueba 3 como su tiempo de corrección no resultan relevantes para el modelo MANOVA propuesto en **Figura 9**.

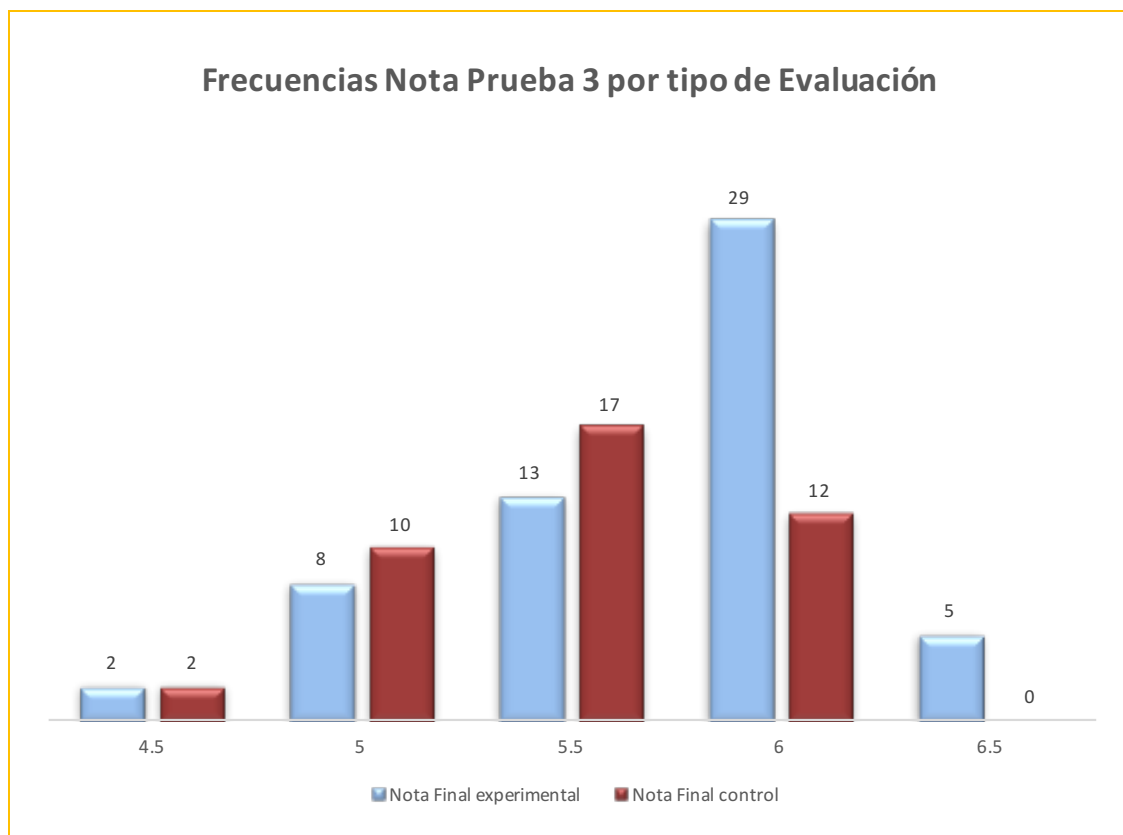


GRÁFICO 7. HISTOGRAMA DE FRECUENCIA DE NOTA PRUEBA 3 POR TIPO DE EVALUACIÓN.

Al observar el **Gráfico 7**, se observa un desplazamiento hacia la derecha, lo que indica un rendimiento mayor. Cabe destacar que se eliminaron de los datos alumnos con promedio final 1.0, dicho alumnos abandonaron el curso de manera temprana, pero no lograron desinscribir de su malla, por lo cual, para efectos de registro académico están considerados. Por lo tanto, para curso control y experimental ningún alumno, que llevo a cabo el curso en su totalidad, reprobó. Sin embargo, se observa un mayor rendimiento en nota final del ramo en experimental versus control.

En **Tabla 18**, se muestran los resultados obtenidos de la prueba de medias correspondiente a nota final con el fin de explorar si las observaciones señaladas son estadísticamente significativas.

TABLA 18. PRUEBAS T-STUDENT PARA NOTA FINAL.

	Prueba de Levene		Prueba t			
	F	Sig.	t	Sig.	Inferior	Superior
Se asumen varianzas iguales	0,898	0,346	-2,725	0,008	-0,638	-0,101
No se asumen varianzas iguales			-2,788	0,006	-0,632	-0,106

Las varianzas de las muestras se asumen igual, no fue posible rechazar la hipótesis nula de la prueba de Levene. Prosiguiendo con la prueba t, esta verifica la observación realizada previamente, por lo cual, existe evidencia estadísticamente significativa para afirmar que la media de grupo experimental versus control no es la misma, y en este caso, es mayor para experimental que control.

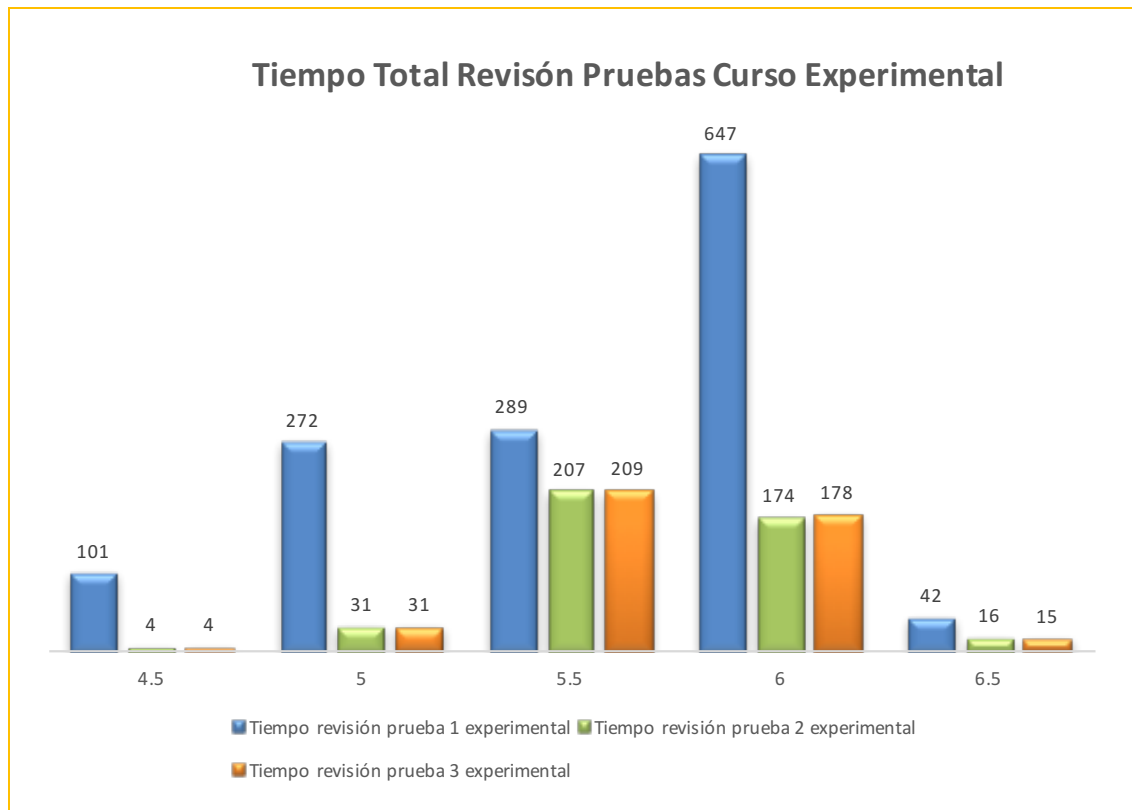


GRÁFICO 8. HISTOGRAMA DE TIEMPOS TOTALES DE REVISIÓN DE NOTA PRUEBA 1, 2 Y 3 POR CURSO EXPERIMENTAL. TIEMPO MEDIDO EN MINUTOS.

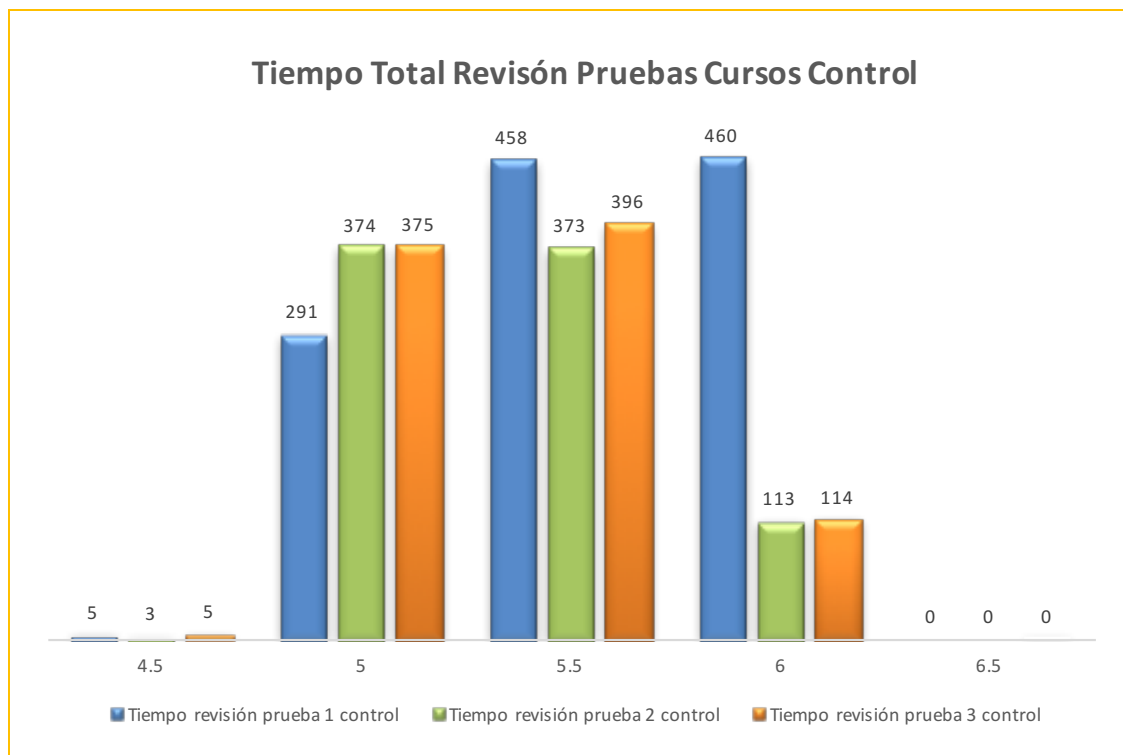


GRÁFICO 9. HISTOGRAMA DE TIEMPOS TOTALES DE REVISIÓN DE NOTA PRUEBA 1, 2 Y 3 POR CURSO CONTROL. TIEMPO MEDIDO EN MINUTOS.

Para complementar un análisis sobre la nota final obtenida, es necesario observar el **Gráfico 8** y **Gráfico 9**, histogramas de tiempos de revisión por parte del curso experimental y los cursos control para las tres evaluaciones. Los tiempos están ordenados por la nota final obtenida, por ejemplo, 375 minutos en **Gráfico 9** sobre tiempo de revisión prueba 1 en nivel 5, corresponde al total de tiempo que revisaron los 10 alumnos presentados en Gráfico 5 en el mismo nivel 5 de calificación. De este modo, permite comparar cuánto tiempo dedicaron a revisar cada evaluación haciendo la correlación con la nota final obtenida del curso.

Resulta interesante analizar que para cursos control los tiempos son relativamente parejos, excepto en prueba 2 y 3, y además son elevados, estamos hablando de más de seis horas para los 10 alumnos que se mencionan en el párrafo anterior. Esto es posible explicarlo por un alumno que debe revisar y volver a revisar su prueba con el objetivo de inferir en que se equivocó, que le faltó entre lo que él respondió y lo que se

esperaba que fuera el rendimiento óptimo. Dicha actividad, requiere de mucho trabajo por parte del estudiante, el cual es posible observar tanto en **Gráfico 9** y en **Gráfico 8** en la prueba 1, dado que esta evaluación también utilizó Emarking normal sin ningún agregado.

Al observar el **Gráfico 8**, se aprecia una gran diferencia entre prueba 1 y las dos pruebas restantes en el tiempo total utilizado por el estudiante para revisión. En contraparte con lo ocurrido en cursos control, el ayudante al generar una mejor retroalimentación el estudiante requiere de un menor tiempo para revisar y procesar su prueba, los mensajes son claros, precisos y dirigidos para él.

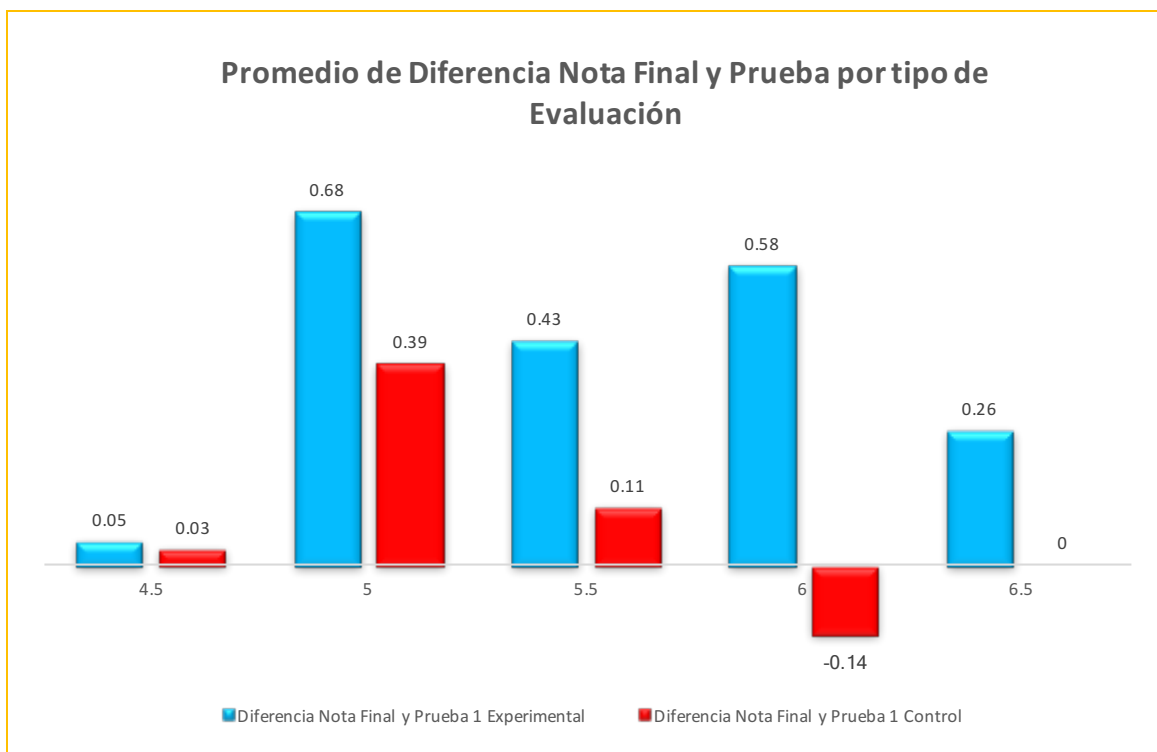


GRÁFICO 10. HISTOGRAMA DE DIFERENCIA PROMEDIO ENTRE NOTA FINAL Y PRUEBA 1 POR TIPO DE EVALUACIÓN.

Al observar el **Gráfico 10** y la media de la variable Diferencia entre nota final y prueba 1 descrita en **Tabla 11**, donde en esta última se reporta la media para cursos control fue de 0,071 y para experimental de 0,499. Esta variable, que es combinación lineal de

otras dos variables, mide el rendimiento del alumno desde la prueba 1 hasta la nota final, en este caso, claramente el aumento más significativo ocurrió en el curso experimental. Además, el Gráfico 10 presenta las diferencias ocurridas entre nota final y prueba uno, ubicando los valores respecto a la clasificación de nota final obtenida, por ejemplo, 0,26 ubicado en nivel 6,5 de **Gráfico 10** corresponden a los 5 alumnos presentes en el **Gráfico 7** en el mismo nivel 6,5.

Es posible concluir que el rendimiento aumentó en los cursos control y experimentales, pero el curso experimental aumentó en un rendimiento mucho más elevado y además estadísticamente significativo. Además, contando con menores tiempos de revisión de prueba que son explicados por el trabajo de los correctores experimentales en la dedicación para crear la retroalimentación.

5.3.3 Conclusiones

Si es posible concluir que el uso del widget con conexión a recursos OER ayuda a los alumnos del curso experimental de sistemas de información con resultados estadísticamente relevantes, esto debido a la información y resultados expuestos en la **sección 5.3.2**

Por una parte, es posible observar menores tiempos de revisión de pruebas por parte de alumnos del curso experimental, rondando valores aceptables, por ejemplo 2,96 horas para 11 alumnos, es decir, 16 minutos por alumnos. En contraparte para los cursos control los tiempos observados son 37,5 minutos por alumno. En ningún caso, un mayor tiempo de revisión significaría un mejor rendimiento, al contrario, un menor tiempo significa mejor trabajo de inferencia para el alumno, un mejor trabajo tratando de encontrar en que se equivocó, si es que no existe retroalimentación o esta es deficiente. En este punto, el widget con conexión a recursos OER cumplió su objetivo bajando la carga de trabajo al estudiante, a tiempo razonables, y permitiendo

comprender en que se había equivocado, cuál era el objetivo de la pregunta y como mejorar a futuro en dicho tópico.

Apoyando la conclusión anterior, tenemos la información presentada por el **Gráfico 7** y la **Tabla 18**, donde es posible diferenciar claramente el rendimiento que obtuvo cada curso. Si observamos los resultados sobre rendimiento del experimento llevado a cabo en programación, hay diferencias menores entre ellas. En cambio, las diferencias presentes en el curso de sistemas son amplias y estadísticamente relevantes, por lo cual, aislando las demás variables la única diferencias y, por ende, la única variable independiente que podría influir en este comportamiento es el tipo de evaluación, que a su vez, es diferenciada por el uso del widget, presentado en **sección 2.4**, que entrega herramientas para que el corrector genere una mejor retroalimentación con el fin de que el alumno la consuma y le permita mejorar su rendimiento. Por lo cual, los resultados apuntan a que la retroalimentación fue consuma un tiempo prudente y de la mano el aumento de rendimiento fue enorme.

Discusión

En el **Capítulo 5**, se analizó en extenso todo el proceso realizado durante el experimento en los cursos de programación y sistemas de información. Antes de llegar a una conclusión final, es importante enfocarse en la validez del análisis e interpretación de ellos, tomando atención a los hechos descritos en la **sección 5.1**. De este modo, la interpretación de los datos se enfocó en aterrizar los resultados al contexto de la universidad, características de los alumnos e hitos ocurridos durante el semestre, con el fin de entregar observaciones completas y válidas.

En la literatura previa no existen investigaciones específicas en el contexto de la retroalimentación y el uso de open educational resources para el apoyo en la creación de retroalimentación y su efecto en el aprendizaje y rendimiento de los estudiantes. Esta da aún más importancia a los resultados obtenidos, ya que representan los primeros pasos en investigaciones que vinculan la educación y la tecnología en la dimensión descrita.

Investigaciones en esta área aportan enormemente a la comunidad científica y, eventualmente, a entidades gubernamentales para el desarrollo de la educación mediante el uso de las Tecnologías de Información. Actualmente, se lleva a cabo un proyecto, FONDEF IT15I10002, financiado por el gobierno de Chile que investiga sobre el impacto del uso de Emarking en la educación básica. El proyecto lleva el nombre de *Enseñanza y evaluación de la escritura mediante una plataforma tecnológica colaborativa*, y es llevado a cabo por la Universidad de Chile.

Es de vital importancia que investigaciones de este tipo lleguen a la comunidad científica internacional, por lo cual es, importante realizar publicaciones en idioma inglés para la correcta difusión. En esta línea, como se mencionó previamente, se publicó en la conferencia IEEE ICALT 2017 un paper sobre el impacto que tiene el widget y OER, propuesto en **sección 2.4**, sobre el corrector y la generación de retroalimentación que él genera.

Como proyección de esta investigación, se espera realizar una segunda publicación que abarque el contenido y resultados expuestos en el presente documento, con el fin de impulsar la investigación a la comunidad internacional. Dicha publicación tiene como objetivo ser publicada en IEEE Transactions on Learning Technologies (TLT).

Finalmente, se espera poder replicar el experimento durante el año 2018, abarcando cursos más diversos que utilizan Emarking, escapando de la línea de cursos de Tecnologías de la Información.

Conclusión y recomendaciones

Mediante la investigación se logró estudiar el impacto que tuvo en el rendimiento de los alumnos el recibir, de parte de los correctores, retroalimentación generada con Open Educational Resources. Las fuentes de información y la interpretación en contexto fueron importantes para poder llegar a conclusiones interesantes, haciendo hincapié en el uso de dos grandes fuentes de información, el curso de Programación de primer semestre de Ingeniería Civil y el curso de Sistemas de Información de noveno semestre de Ingeniería Civil, ambos en la Universidad Adolfo Ibáñez.

Para ambos cursos se utilizaron los mismos análisis, la diferencia ocurrió en la construcción del modelo MANOVA, donde como fuente de ingreso se utilizó el mismo set de variables y se fue ajustando mediante su nivel de influencia que, finalmente dio origen a dos modelos **Figura 8** y **Figura 9**, en términos de variables dependientes implicadas, diferentes pero que a la vez trataban de explicar el mismo fenómeno.

Para el curso de programación, se encontró que las variables más importantes para el modelo fueron: *Tiempo Revisión Prueba 1*, *Nota Prueba 1*, *Nota final* y *Diferencia entre Nota Final y Prueba 2*. Se analizó en detalle el comportamiento de cada una de las variables, con un foco siempre en su impacto en el aprendizaje del estudiante, medido mediante el rendimiento que tuvo como nota final del curso.

Respecto al análisis de las cuatro variables más influyentes, en los cursos control el comportamiento de éstas fue mejor evaluado, es decir, las notas de la prueba 1 fueron mejores, así como también la nota final del curso fue mejor para los cursos control, marcando diferencias estadísticamente significativas entre ambos cursos.

La variable *Diferencia entre nota final y prueba 2*, que se definió como el rendimiento que tuvo el estudiante en el curso, resultó ser mejor para el curso control, pero en este caso hubo mejora en el curso experimental.

Como se mencionó en la sección **5.2.3**, no es posible concluir que la mejora en los cursos experimentales, dado que si hubo mejora, es atribuible a la variable de tipo de evaluación, que a su vez diferencia qué curso usó el nuevo widget. No existió evidencia estadística que permitiera asociar el redimiendo de los cursos experimentales al uso del widget con conexión a OER. De este modo, el tratamiento que necesita el curso de programación debe ser personalizado por las dificultades que este plantea a la coordinación, por lo cual, como experiencia para la investigación resulta muy interesante los resultados obtenidos principalmente por el hecho de llevar a cabo un experimento en un ambiente no controlado, es decir, un ambiente completamente real en la universidad, donde pueden ocurrir situaciones que dificultan las labores de profesores y correctores. La importancia recae en medir el comportamiento de la herramienta en este ambiente, donde a futuro se vea desenvuelta, y poder contar hoy con los resultados obtenidos a fin de poder mejorar la experiencia, diseño experimental y procesos.

Para el curso de Sistemas de Información, se encontró que las variables más importantes para el modelo fueron: *Tiempo revisión prueba 3*, *Nota final y Diferencia nota final y prueba 1*. Los resultados del análisis a las tres variables más influyentes distan mucho de lo que se apreció previamente respecto al curso de Programación. En el caso de Sistemas de Información, el rendimiento inicial en la prueba 1 fue de 5,1 para los cursos control y de 4,9 para el curso experimental. En la instancia de prueba 1, los cursos control y experimental utilizaron la herramienta Emarking normal, aún no se había agregado al curso experimental el widget, por lo cual, se quería medir una base para luego ser contrastada y determinar el aprendizaje de los alumnos mediante el rendimiento. Es en este punto, donde hay una inflexión en los datos del curso

experimental y el rendimiento crece a un ritmo significativo, es más, si analizamos el **Gráfico 10** en ningún momento la diferencia entre nota final y prueba 1 fue negativa para el curso experimental, y más aún, es estadísticamente mayor que la diferencia presentada por los cursos control.

Por otro lado, al analizar los tiempos en los cuales los alumnos estaban revisando sus pruebas, que ejemplifica la calidad de la retroalimentación y mensajes dejados por el corrector para ellos. Los tiempos, que se puede apreciar en el Gráfico 8 y Gráfico 9 son mucho mayores para los cursos control que para el curso experimental, lo que nos permite inferir que el estudiante tuvo que gastar mucho más tiempo en descubrir sus inquietudes, en encontrar cual era la respuesta correcta para luego de ello tener una lectura clara de la evaluación. De este modo, un mayor tiempo de revisión de la prueba solo apunta a una menor calidad de la retroalimentación. Como se mencionó en **sección 5.3.3**, un alumno de curso experimental pasó aproximadamente 16 minutos revisando una evaluación mientras que un alumno de curso control pasó aproximadamente 37,5 minutos revisando su prueba.

Por lo tanto, existe evidencia estadística significativa suficiente, aislando las variables dependientes y, trabajando con la variable independiente de tipo de evaluación, para no lograr rechazar la hipótesis de investigación, por lo cual, el aumento del rendimiento de los alumnos del curso experimental es debido al uso del widget con conexión a OER por parte de los correctores.

En síntesis, las experiencias analizadas tanto para el curso de Programación como para el curso de Sistemas de Información fueron satisfactorias. En primer lugar, porque investigaciones en la línea de retroalimentación usando open educational resources no existen en la literatura, en segundo lugar, porque los datos fueron obtenidos desde un contexto real en la universidad, no fue un experimento moldeado donde cada variable estuviera controlada y, en ese sentido realizar un experimento donde se sabe con 100%

de exactitud el resultado obtenido, no tiene valor para la comunidad científica y carece de peso investigativo.

Por último, se logró obtener resultados satisfactorios en los espacios donde se realizó el experimento, en el curso de Programación fue posible entender su contexto y dificultad, donde los resultados no fueron los esperados respecto al impacto del widget en el rendimiento de los alumnos, pero completamente válidos y analizables. En el curso de Sistemas de Información los resultados fueron los esperados respecto al impacto que generó la retroalimentación creada con el widget, con un efecto no esperado durante la fase de diseño experimental, que fue la disminución de los tiempos de revisión por parte de los alumnos y, tiene mucho sentido debido a que para el alumno es mucho más fácil y transparente revisar una evaluación de este tipo, sin necesidad de perder tiempo en inferir en qué fallo o como debería haber contestado.

Referencias

- [1] Albertson, L. M. (1986). Personalized feedback and cognitive achievement in computer-assisted instruction. *Journal of Instructional Psychology*.
- [2] Azevedo, R., & Bernard, R. M. (1995). A meta-analysis of the effects of feedback in computer-based instruction. *Journal of Educational Computing Research*, 13(2), 111-127.
- [3] Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: principles, policy & practice*, 5(1), 7-74.
- [4] Boud, D., & Falchikov, N. (2006). Aligning assessment with long-term learning. *Assessment and Assessment in Higher Education*, 31, 399–413.
- [5] Cohen, V. B. (1985). A reexamination of feedback in computer-based instruction: Implications for instructional design. *Educational Technology*, 25(1), 33–37
- [6] Eurant, Michael. (2006). Feedback. *Learning in Health and Social Care*, 5, 111–118.
- [7] Evans, C. (2013). Making Sense of Assessment Feedback in Higher Education. *Review of Educational Research*, 83(1), 70-120.
- [8] Higgins, R., Richard, H., Peter, H., & Alan, S. (2001). Getting the Message Across: The problem of communicating assessment feedback. *Teaching in Higher Education*, 6(2), 269–274.
- [9] Hyland, P. (2000). Learning from feedback on assessment. *The practice of university history teaching*, 233-247.
- [10] Jeria, Hans; Villalon, Jorge. Incorporating Open Education Resources into Computer Supported Marking Tool to Enhance Formative Feedback Creation. *En Advanced Learning Technologies (ICALT), 2017 IEEE 17th International Conference on*. IEEE, 2017. p. 256-260.
- [11] Kulhavy, R. W., & Stock, W. A. (1989). Feedback in written instruction: The place

- of response certitude. *Educational Psychology Review*, 1(4), 279-308.
- [12] Lin, X., Hmelo, C., Kinzer, C.K., & Secules, T.J. (1999). Designing technology to support reflection. *Educational Technology Research and Development*, 47(3), 43-62.
 - [13] Matas, M., & Villalon, J. (2015, July). Effects of group and task size in Computer Supported Collaborative Marking. In *Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on* (pp. 228-232). IEEE.
 - [14] Narciss, S., & Huth, K. (2004). How to design informative tutoring feedback for multimedia learning. *Instructional design for multimedia learning*, 181-195.
 - [15] Nicol, D. J., & Debra, M. (2006). Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2), 199–218.
 - [16] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
 - [17] Price, M., Margaret, P., Karen, H., Jill, M., & Berry, O. 'donovan. (2010). Feedback: all that effort, but what is the effect? *Assessment & Assessment in Higher Education*, 35(3), 277–289.
 - [18] Templeton, Gary F. A Two-Step Approach for Transforming Continuous Variables to Normal: Implications and Recommendations for IS Research. *CAIS*, 2011, vol. 28, p. 4.
 - [19] Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, 78(1), 153-189.
 - [20] Villalon, J., (2012). An eMarking Tool for Paper Based Assessments. In 2012 IEEE 12th International Conference on Advanced Learning Technologies. <http://doi.org/10.1109/icalt.2012.221>
 - [21] Villalon, J., Beghelli, A., & Merino, J. (2016, July). Cost and benefit of a computer supported Real Time Delphi method on inter-rater reliability using rubrics in highly subjective assessment tasks. In *Advanced Learning Technologies (ICALT), 2016*

IEEE 16th International Conference on (pp. 202-206). IEEE.

- [22] Yuan, L., MacNeill, S., & Kraan, W. G. (2008). Open Educational Resources - opportunities and challenges for higher education, 34.

ANEXOS

A Paper publicado en conferencia en el marco del MCI

2017 IEEE 17th International Conference on Advanced Learning Technologies

Incorporating Open Education Resources into Computer Supported Marking Tool to enhance formative feedback creation

Hans Jeria

Facultad de Ingenieria y Ciencias
Universidad Adolfo Ibanez
Santiago, Chile
hansjeria@gmail.com

Jorge Villalon

Facultad de Economia y Empresa
Universidad Diego Portales
Santiago, Chile
jorge.villalon@udp.cl

Abstract- The importance of formative feedback in higher education institutions is out of doubt, however, reports on poor feedback from students and a high cost in producing quality feedback from teachers demand a different line of action. This article proposes the use of Open Education Resources as suggested content for markers in order to enhance their provided feedback. An evaluation was performed using a computer supported marking tool which connect the mark activity with the OER to encourage the marker to produce formative feedback while she is marking exams. Control and experimental groups of markers were arranged, with the former using only a text box to include commentaries as feedback and the latter the OER connected search tool. Both groups marked the same set of exams after a training session on quality feedback. Results showed that the use of OER in the interface to provide feedback made markers generate more and longer feedback than those in the control group.

electronic feedback; open educational resources; emarking; formative feedback;

I. INTRODUCTION

Feedback used in educational contexts is considered as crucial to improve knowledge and skill acquisition. Shute defines formative feedback as: "...information communicated to the learner that is intended to modify his or her thinking or behavior for the purpose of improving learning" [14]. Intentional or non-intentional formative feedback in the first years of higher education is particularly crucial for students, as it does not only play a vital role in the creation of their learning but it can also influence their professional identity. [1] In this way, the use of feedback can convert an assessment into an instrument for developing learning. [2], and even more significantly, feedback can improve the depth of learning. [3]

For feedback to reach its full potential, several aspects of its quality have been studied, which were summarized by Nicol & Debra [6] in seven principles of quality feedback. Principle 1 states that quality feedback must "help clarify what good performance is", meaning that feedback should be able to explain the objectives or specific goals for each assessment question and the expected level of performance. Principles 2 and 3, "Facilitate self-assessment" and "Deliver high quality feedback information" are related to information that helps students self-regulate. Such characteristics can be found in studies that focus on feedback from the students' point of view

[9], which clarify that quality feedback allows the student to compare their current performance and optimal performance, and moreover allow the student to understand what to do to reduce the gap between their performance and the optimal performance.

Despite the recognized benefits that quality formative feedback has, there are still many challenges in the assurance of quality feedback, as its development requires an important effort. Studies that report on the quality of feedback from the student point of view, have found that a wide majority consider feedback as hard to understand, including impersonal and standardized comments, which is perceived as negative and lacking information about the student own response [5]. Such findings are in line with Boud and Falchicov, whom identified that personal and institutional factors affect the way students experience and consume feedback [8], meaning that no common standard can be found in higher education scenarios. Prince et. al. proposed that feedback's potential can only be reached if students are trained on how interpret the feedback, to make connections between the work performed, the corresponding feedback and ways of improving in the future [7]. However, Evans raises concerns on the need for training students on how to understand feedback, as care must be taken when providing very explicit instructions or guidelines on the interpretation of feedback, since there is an inherent danger of explicitly guiding the student, which can create a dependency, and limit thinking [10].

Technology to support the process of providing formative feedback can help tackle the problem of raising costs for teachers and tutors. Technology can help reduce these costs by allowing feedback to be created collaboratively, reusing feedback within evaluations and markers, and simplifying its delivery to students among others. However, all these features are domain independent, and cannot benefit from domain specific sources, that can help students expand their knowledge base, such as Open Education Resources (OER).

We propose that OER can be an excellent source for quality information in formative feedback, and that technology can facilitate its use within evaluation processes, helping to create awareness of such repositories within lecturers and students. This article reports on the design and implementation of a search widget, which is embedded in the user interface of an On-Screen Marking tool that supports the creation of formative feedback. The search feature allows the discovery of learning materials from OERs and their use in the

feedback creation by teachers and tutors. This study aims to answer the following research question: Does the presence of search results from OERs affect the comments provided by tutors within a marking process?

Section II provides the theoretical background that supports the design of the widget, section III presents the details on the design of the widget, section IV explains the experimental setup for the evaluation, section V discusses the results from the experiments and section VI concludes.

II. PREVIOUS WORK

Quality formative feedback

Seven principles of good feedback practice have been proposed by Nicol & Debra, they defined that quality feedback is the one that:

1. helps clarify what good performance is (goals, criteria, expected standards);
2. facilitates the development of reflection and self-assessment in learning;
3. delivers high quality information to students about their learning;
4. encourages teacher and peer dialogue around learning;
5. encourages positive motivational beliefs and self-esteem;
6. provides opportunities to close the gap between current and desired performance;
7. provides information to teachers that can be used to help shape the teaching.

Even though such a definition seems to be widely accepted, Shute argues that there is no consistent pattern in the results and many contradictory results in the studies on formative feedback [14]. According to her, the problem resides on the difference in importance that researchers give to different principles. Accordingly, as [15], [16] and [17] mention, the main objective of formative feedback, whether delivered by a teacher or computer, in the classroom or anywhere else, is to improve learning and/or performance, leading to precise guidance, pointing to concepts and skills. This objective will only be fulfilled if the results are correctly delivered to the student or apprentice. In the proposed tool, there is a focus on two principles of quality feedback, which are reported from students to be a problem in actual scenarios: Providing high quality information about their learning while clarifying what good performance is. The goal is to support the creation of feedback once the teacher identifies a performance level in a student answer, and intervene the moment in which the judgment is made by proposing educational resources that may help students develop their performance in such domain.

More inherent characteristics of feedback have been proposed, according to Black and Wiliam [18], feedback has two main functions: directive and facilitative. Directive feedback informs the student of what they have to fix or revise. This function tends to be more specific than the facilitative function, which provides comments and suggestions to help guide students in their own review and conceptualization. Furthermore, from its information

perspective, effective feedback provides students with two types of information: verification and elaboration [19]. Verification is defined as the simple judgment of whether an answer is correct or not, and elaboration is the informative aspect of the message, which provides clues to guide the student towards a correct answer. In the design of the tool the focus is on the facilitative function, as it is already integrated into an On-Screen marking tool which includes rubrics, that provide a structured base for judgments on the evaluated work. From the information perspective, the purpose is to support elaboration, as extra information lies beyond the correctness of the answer, and allows precisely to guide the student towards a wider knowledge.

Open Educational Resources

Seymour Papert proposed in Mindstorms that it is possible to modularize knowledge through technology, as when knowledge can be broken up into mind-size bites, it will be more communicable, digestible and easy to build [11]. From Papert's ideas, presented more than 35 years ago, it is now possible to find such modularity in the Open Educational Resources (OER) format, which does not have territorial or economic restrictions. The number of available OERs has been progressively increasing in recent years, and according to the Organization for Economic Cooperation and Development (OECD), there are more than 3,000 open access courses and more than 300 universities around the world sharing content [12].

The open access courses or OpenCourseWare (OCW) possess content with intellectual property licenses ensuring the relinquishment of copyright, meaning that they can be reused freely when the author is cited. Typical examples are MIT's Open Course Ware, Open Resource for Education Consortium, Japanese OCW Consortium and ParisTech OCW Project.

As OERs' modularity comes from the need for its reuse, they become excellent sources for enhancing feedback, by being self-contained explanations of specific domains that facilitate their understanding, and by connecting to other resources and therefore opening a window to students to a wider source of domain knowledge for their own reflection.

On Screen Marking tools

Emarking is the base technology used for on-screen marking in our experiments, reported in [13], [20] and [21]. Previous studies have explored how the tool can support inter-rater agreement and improve markers' performance on evaluations providing specific features such as markers' collaboration, automatic calculation of inter-rater agreement, and facilitating feedback delivery to students among others.

One drawback of the system is that it supported marking processes only from a procedural perspective and never from a domain perspective. This proposal is the first attempt to bring domain-specific information to help improve the quality of formative feedback. The tool allows to scan handwritten exams and uses rubrics to provide a structured way to compare

judgments while using criteria as a minimum level of feedback.

III. EMARKING ENHANCED FEEDBACK FEATURES

Modifications were made to Emarking [13] to include the new feature for enhanced feedback. Firstly, a new field was added in the creation of an Emarking activity (a marking session), as shown in Figure 1, which allows key-words or phrases to be used as queries to find OER material, which can subsequently be made accessible to the marker when grading and generating feedback for the student.

In the original interface, when a mark is given by a teacher or tutor, a level within a rubric criterion is selected as the performance achievement of the student. Once selected, a dialog is shown in which the selected level can be changed, the score assigned by the level can be modified using the bonus field, and a comment can be added as written feedback. The dialog is shown in Figure 2; the interface is simple, and verification is facilitated by linking the feedback with a specific level of a rubric criterion. On the other hand, elaboration is aimed to facilitate students to receive information on forming their own review and guide them towards a correct answer. Despite its complexity, the interface only provides a text box to be filled if the maker decides to do so.

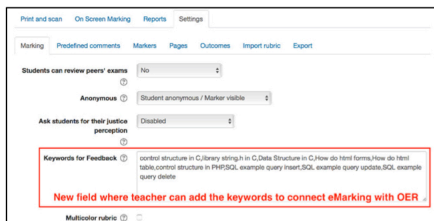


Figure 1. Interface for creating a new Emarking activity.

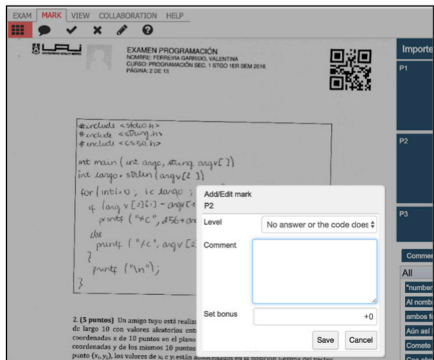


Figure 2. Original interface for feedback.

Figure 3 shows the proposed interface, which includes a right hand panel that provides markers with educational content from three sources: OpenCourseWare from the Massachusetts Institute of Technology (<https://ocw.mit.edu/index.htm>), MERLOT Multimedia Educational Resource for Learning and Online Teaching (<https://www.merlot.org/merlot/index.htm>) and finally the University's LMS (Moodle), which provides access to educational material from the course through the Emarking resource. A second panel on the lower left side, shows resources added by markers, as feedback specifically assigned to the criterion level assigned. This panel includes all previously mentioned characteristics of the original interface: a text box to leave comments and suggestions, a select input to change the rubric criteria selected and a text box to modify the score assigned by the criterion level.

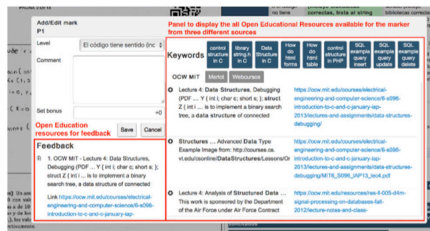


Figure 3. OER enhanced interface for feedback.

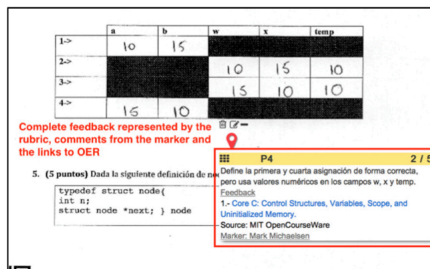


Figure 3. Student view for the correction made by the corrector for a problem.

Finally, Figure 4 shows the student display of the feedback, which was also modified for the OER enriched version. In the new version, the rubric is integrated into the display, which provides verification, comments written by the marker and links to resources. These last two deal with the elaboration component, which is essential for the creation of formative feedback.

In the next two chapters, the research question will be answered, the focus is on how the tool affects feedback produced by markers.

IV. EVALUATION

The experimental setup included hand written final exams taken from a first year's Engineering programming class. A single class of 46 students that were randomly assigned from a cohort of 398 by the admission office was chosen to digitize their answers for the experiment. A group of six markers evaluated the same 46 exams in three separated groups, one that used the normal Emarking tool and the remaining two used the enhanced feedback interface. Each marker had an assignment of 23 exams within a group, and each subgroup was also randomly assigned. The rubric used for marking had sixteen criteria, and was the same one used in marking the for the final grades.

All markers were trained in the feedback creation process and on the theoretical framework that supports the concept of quality formative feedback used for the design of the tool proposed in this article. In addition, all six markers knew and used the On-Screen marking tool before. An extra training session was performed on the use of the enhanced feedback interface for markers in the experimental group.

In the results, the length of the comments written by teachers and tutors was calculated, and the number of links to OERs was also obtained. This latest indicator, number of links to OER, were only available on experimental group which are one of the variable to consider who the presence of search results from OERs affect the creation of feedback by the marker.

V. RESULTS

Table 1 shows a summary of all criteria evaluated for the 46 exams (same rubric with 16 criteria) by six markers (23 exams each). All marks that indicated the maximum level in a rubric were not considered. As mentioned previously, criteria evaluations were removed when the maximum score was assigned, as most of such marks included no written feedback. As it is shown, markers in the experimental group (markers 3 to 6) had an average comment length that was significantly higher than the control group (markers 1 and 2), with the exception of marker 5. These results allow us to confirm the research question that the OER's search tool does affect the quantity of feedback provided by teachers and tutors.

Table 1. Feedback comment length intervals per marker

Comment length	Marker 1	Marker 2	Marker 3	Marker 4	Marker 5	Marker 6
0-20	193	149	70	207	37	49
20-40	8	11	18	26	4	44
40-60	3	22	11	26	8	55
60-80	1	28	7	13	1	54
80-100	0	3	14	2	0	21
100-120	0	0	24	1	2	5
120-140	0	0	8	3	0	0
140-160	0	0	1	0	0	0
160-180	0	0	29	14	0	0
180-200	0	0	1	0	0	0
200-220	0	0	2	0	0	0
220-240	0	0	0	0	0	0
240-260	0	0	10	0	0	0
260-280	0	0	1	0	0	0
AVERAGE	2.5	18.1	77.1	21.2	16.6	45.4

To isolate other factors, a MANOVA analysis was performed to explore the results obtained using comment length as dependent variable, in order to test the hypothesis that markers in the experimental group, this is, that used the enhanced feedback tool generated more and longer feedback, based on comments made, comments' length and the number of links added from OER materials. Results obtained are shown in Table 2. The dependent variable is the comment length and the independent variables are the type of Emarking, marker, score assigned by the marker and number of links to educational materials added.

Table 2. MANOVA analysis for comments' length

	Value
Marker	<0.001
Score	0.0033
Emarking type	<0.001
Number of links	<0.001

From the results shown that the score assigned for the question is the least related variable for the length of feedback. We consider this important, as it shows that quality formative feedback is required at all levels of performance and that the cost of producing such feedback is always important. Another important aspect is that even though all other variables are important predictors for comments' length, comparing those results per marker in table 1, despite two markers with very low averages in their comments' length, it did not make Emarking type less predicting, on the contrary, the p-value for type is slightly lower (one order of magnitude) than the p-value for Marker.

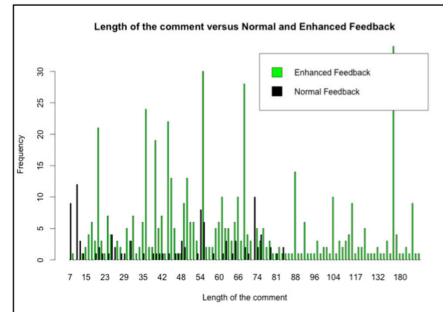


Figure 4. Barplot of feedback commentary length per criteria, grouped by the e Marking type.

VI. CONCLUSIONS AND FUTURE WORK

In this article the idea of using an on-screen marking tool (Emarking) to support markers on providing feedback was designed and evaluated. The search widget was an extension of a rubric based marking system in which every judgment made to the student's object (exam, homework, etc.) had the opportunity to contribute formative feedback. However, despite the opportunity for providing feedback, there was no

standard on the amount of feedback provided in a higher education scenario.

The proposal for the new widget is to include domain specific information to support markers on the creation of enhanced feedback. A set of key-words, defined previously by the author of the evaluation, was shown effective to suggest OER materials to markers, that had the opportunity to add them to their feedback comments. The widget was evaluated using control and experimental groups, that marked the same set of 46 final exams.

Results showed that the length of the written feedback left by markers is directly affected when the widget is provided or not, results were statistically significant. They also showed that feedback length is also highly dependent on the number of links added from the suggested feedback, which can be interpreted as a need to explain such inclusion within the feedback from the markers' perspective. Thereby, it is not possible to refuse the hypothesis, consequently the hypothesis is accepted.

In summary, results show that enhanced feedback can be supported with technology, and therefore the design of such tools can become an important part of a higher education system in which the number of students continues to grow.

Future work in this area includes more experimental designs, including instruments to measure students' perceptions on important aspects from feedback. Another aspect that can reduce the cost of preparing Emarking supported marking processes would be the automatic extraction of a set of key-words from the course material or the exams' text and rubric. Also, measuring the impact on students, either on performance and/or learning, which has the mentioned widget.

Further, it is necessary more studies in the field of formative feedback with the goal to provide tools for the marker, which one impact directly to students.

Finally, it is important to mention that in these experiments no collaboration was encouraged from the research team, therefore a collaborative version of the widget may reduce even more the cost of producing feedback, leaving more time for markers on providing personalized comments.

ACKNOWLEDGMENTS

Financial support from FONDECYT project 11130055 is gratefully acknowledged. We also acknowledge the effort of Nicolas Herrera and Roberto Jaunez.

VII. REFERENCES

- [1] Eurant, Michael. (2006). Feedback. Learning in Health and Social Care, 5, 111-118.
- [2] Hyland, P. (2000). Learning from feedback on assessment. The practice of university history teaching, 233-247.
- [3] Lin, X., Hmelo, C., Kinzer, C.K., & Secules, T.J. (1999). Designing technology to support reflection. *Educational Technology Research and Development*, 47(3), 43-62.
- [4] Panitz, T. (1999). Collaborative versus cooperative learning: A comparison of the two concepts which will help us understand the underlying nature of interactive learning.
- [5] Higgins, R., Richard, H., Peter, H., & Alan, S. (2001). Getting the Message Across: The problem of communicating assessment feedback. *Teaching in Higher Education*, 6(2), 269-274.
- [6] Nicol, D. J., & Debra, M. (2006). Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2), 199-218.
- [7] Price, M., Margaret, P., Karen, H., Jill, M., & Berry, O. 'donovan. (2010). Feedback: all that effort, but what is the effect? *Assessment & Assessment in Higher Education*, 35(3), 277-289.
- [8] Boud, D., & Falchikov, N. (2006). Aligning assessment with long-term learning. *Assessment and Assessment in Higher Education*, 31, 399-413.
- [9] Sadler, D. R., & Royce Sadler, D. (1989). Formative assessment and the design of instructional systems. *Instructional Science*, 18(2), 119-144.
- [10] Evans, C. (2013). Making Sense of Assessment Feedback in Higher Education. *Review of Educational Research*, 83(1), 70-120.
- [11] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- [12] Yuan, L., MacNeill, S., & Kraan, W. G. (2008). Open Educational Resources - opportunities and challenges for higher education, 34.
- [13] Villalon, J., (2012). An eMarking Tool for Paper Based Assessments. In 2012 IEEE 12th International Conference on Advanced Learning Technologies. <http://doi.org/10.1109/icalt.2012.221>
- [14] Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, 78(1), 153-189.
- [15] Albertson, L. M. (1986). Personalized feedback and cognitive achievement in computer-assisted instruction. *Journal of Instructional Psychology*.
- [16] Azevedo, R., & Bernard, R. M. (1995). A meta-analysis of the effects of feedback in computer-based instruction. *Journal of Educational Computing Research*, 13(2), 111-127.
- [17] Narciss, S., & Huth, K. (2004). How to design informative tutoring feedback for multimedia learning. *Instructional design for multimedia learning*, 181-195.
- [18] Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: principles, policy & practice*, 5(1), 7-74.
- [19] Kulhavy, R. W., & Stock, W. A. (1989). Feedback in written instruction: The place of response certitude. *Educational Psychology Review*, 1(4), 279-308.
- [20] Matas, M., & Villalon, J. (2015, July). Effects of group and task size in Computer Supported Collaborative Marking. In *Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on* (pp. 228-232). IEEE.
- [21] Villalon, J., Beghelli, A., & Merino, J. (2016, July). Cost and benefit of a computer supported Real Time Delphi method on inter-rater reliability using rubrics in highly subjective assessment tasks. In *Advanced Learning Technologies (ICALT), 2016 IEEE 16th International Conference on* (pp. 202-206). IEEE.

B Marco de trabajo propuesto por Valerie Shute

Table 2

Formative Feedback Guidelines to Enhance Learning (Things to Do)

	Prescription	Description and references
1	Focus feedback on the task, not the learner.	Feedback to the learner should address specific features of the learner's work in relation to the task, with suggestions on how to improve (e.g., Butler, 1987; Corbett & Anderson, 2001; Kluger & DeNisi, 1996; Narciss & Huth, 2004).
2	Provide elaborated feedback to enhance learning.	Feedback should describe the what, how, and/or why of a given problem. This type of cognitive feedback is typically more effective than verification of results (e.g., Bangert-Drowns et al., 1991; Gilman, 1969; Mason & Bruning, 2001; Narciss & Huth, 2004; Shute, 2006).
3	Present elaborated feedback in manageable units.	Provide elaborated feedback in small enough pieces so that it is not overwhelming and/or discarded (Bransford et al., 2000; Sweller et al., 1998). Presenting too much information may not only result in a superficial learning, but also invoke cognitive overload (e.g., Mayer & Moreno, 2002; Phye & Bender, 1989). A stepwise presentation of feedback offers the possibility to control for mistakes and gives learners sufficient information to correct errors on their own.
4	Be specific and clear with feedback messages.	If feedback is not specific or clear, it can impede learning and can frustrate learners (e.g., Moreno, 2004; Williams, 1997). If possible, try to link feedback clearly and specifically to goals and performance (Hoska, 1993; Song & Keller, 2001).
5	Keep feedback as simple as possible but no simpler (based on learner needs and instructional constraints).	Simple feedback is generally based on one cue (e.g., verification or hint) and complex feedback on multiple cues (e.g., verification, correct response, error analysis). Keep feedback as simple and focused as possible. Generate only enough information to help students and not more. Kulhavy et al. (1985) found that feedback that was too complex did not promote learning compared to simpler feedback.
6	Reduce uncertainty between performance and goals.	Formative feedback should clarify goals and seek to reduce or remove uncertainty in relation to how well learners are performing on a task and what needs to be accomplished to attain the goal(s) (e.g., Ashford, Blatt, & VandeWalle, 2003; Bangert-Drowns et al., 1991).
7	Give unbiased, objective feedback, written or via computer.	Feedback from a trustworthy source will be considered more seriously than other feedback, which may be disregarded. This may explain why computer-based feedback is often better than human-delivered in some experiments in that perceived biases are eliminated (see Kluger & DeNisi, 1996).
8	Promote a learning goal orientation via feedback.	Formative feedback can be used to alter goal orientation—from a focus on performance to a focus on learning (Hoska, 1993). This can be facilitated by crafting feedback emphasizing that effort yields increased learning and performance and that mistakes are an important part of the learning process (Dweck, 1986).
9	Provide feedback after learners have attempted a solution.	Do not let learners see answers before trying to solve a problem on their own (i.e., presearch availability). Several studies that have controlled presearch availability show a benefit of feedback, while studies without such control show inconsistent results (Bangert-Drowns et al., 1991).

Table 3***Formative Feedback Guidelines to Enhance Learning (Things to Avoid)***

	Prescription	Description and references
10	Do not give normative comparisons.	Feedback should avoid comparisons with other students—directly or indirectly (e.g., grading on the curve). In general, do not draw attention to self during the course of learning (Kluger & DeNisi, 1996; Wiliam, in press).
11	Be cautious about providing overall grades.	Feedback should note areas of strength and provide information on how to improve, as warranted and without overall grading. Wiliam (in press) summarized the following findings: (a) students receiving just grades showed no learning gains, (b) those getting just comments showed large gains, and (c) those with grades and comments showed no gains (likely due to focusing on the grade and ignoring comments). Effective feedback relates to the content of the comments (Butler, 1987; McColskey & Leary, 1985).
12	Do not present feedback that discourages the learner or threatens the learner's self-esteem.	This prescription is based not only on common sense, but also research reported in Kluger and DeNisi (1996), which reports feedback interventions that undermine learning as it draws focus to the self and away from the task at hand. In addition, do not provide feedback that is either too controlling or critical of the learner (Baron, 1993; Fedor, Davis, Maslyn, & Mathieson, 2001).
13	Use praise sparingly, if at all.	Kluger & DeNisi (1996), Butler (1987), and others have noted that use of praise as feedback directs the learner's attention to self, which distracts from the task and consequently from learning.
14	Try to avoid delivering feedback orally.	This also was addressed in Kluger & DeNisi (1996). When feedback is delivered in a more neutral manner (e.g., written or computer-delivered), it is construed as less biased.
15	Do not interrupt the learner with feedback if the learner is actively engaged.	Interrupting a student who is immersed in a task—trying to solve a problem or task—can be disruptive to the student and impede learning (Corno & Snow, 1986).
16	Avoid using progressive hints that always terminate with the correct answer.	While hints can be facilitative, they can also be abused. If they are employed to scaffold learners, make provisions to prevent their abuse (e.g., Aleven & Koedinger, 2000; Shute, Woltz, & Regian, 1989). Consider using prompts and cues (i.e., more specific kinds of hints).
17	Do not limit the mode of feedback presentation to text.	Exploit the potential of multimedia to avoid cognitive overload due to modality effects (e.g., Mayer & Moreno, 2002). Do not default to presenting feedback messages as text. Instead, consider alternative modes of presentation (e.g., acoustic, visual).
18	Minimize use of extensive error analyses and diagnosis.	The cost of conducting extensive error analyses and cognitive diagnosis may not provide sufficient benefit to learning (Sleeman et al., 1989; VanLehn et al., 2005). Furthermore, error analyses are rarely complete and not always accurate, thus only are helpful in a subset of circumstances.

Table 4***Formative Feedback Guidelines in Relation to Timing Issues***

	Prescription	Description and references
19	Design timing of feedback to align with desired outcome.	Feedback can be delivered (or obtained) either immediately after some activity or delayed. Immediate feedback can help fix errors in real-time, producing greater immediate gains and more efficient learning, (Corbett & Anderson, 2001; Mason & Bruning, 2001), but delayed feedback has been associated with better transfer of learning (e.g., Schroth, 1992).
20	For difficult tasks, use immediate feedback.	When a student is learning a difficult new task (where difficult is relative to the learner's capabilities), it is better to use immediate feedback, at least initially (Clariana, 1990). This provides a helpful safety net so the learner does not get bogged down and/or frustrated (Knoblauch & Brannon, 1981).
21	For relatively simple tasks, use delayed feedback.	When a student is learning a relatively simple task (again, relative to capabilities), it is better to delay feedback to prevent feelings of feedback intrusion and possibly annoyance (Clariana, 1990; Corno & Snow, 1986).
22	For retention of procedural or conceptual knowledge, use immediate feedback.	In general, there is wide support for use of immediate feedback to promote learning and performance on verbal, procedural, and even tasks requiring motor skills (Anderson et al., 2001; Azevedo & Bernard, 1995; Corbett & Anderson, 1989, 2001; Dihoff et al., 2003; Phye & Andre, 1989).
23	To promote transfer of learning, consider using delayed feedback.	According to some researchers (e.g., Kulhavy et al., 1985; Schroth, 1992), delayed may be better than immediate feedback for transfer task performance, although initial learning time may be depressed. This needs more research.

Table 5***Formative Feedback Guidelines in Relation to Learner Characteristics***

	Prescription	Description and references
24	For high-achieving learners, consider using delayed feedback.	Similar to the Clariana (1990) findings cited in Table 4, high-achieving students may construe a moderate or difficult task as relatively easy and hence benefit by delayed feedback (see also Gaynor, 1981; Roper, 1977).
25	For low-achieving learners, use immediate feedback.	The argument for low-achieving students is similar to the one above, only these students need the support of immediate feedback in learning new tasks they may find difficult (see Gaynor, 1981; Mason & Bruning, 2001; Roper, 1977).
26	For low-achieving learners, use directive (or corrective) feedback.	Novices, or struggling students, need support and explicit guidance during the learning process (Knoblauch & Brannon, 1981; Moreno, 2004), thus hints may not be as helpful as more explicit, directive feedback.
27	For high-achieving learners, use facilitative feedback.	Similar to the above, high-achieving students or more motivated ones benefit from feedback that challenges them, such as hints, cues, and prompts (Vygotsky, 1987).
28	For low-achieving learners, use scaffolding.	Provide early support and structure for low-achieving students (or those with low self-efficacy) to improve learning and performance (e.g., Collins et al., 1989; Graesser, McNamara, & VanLehn, 2005).
29	For high-achieving learners, verification feedback may be sufficient.	Hanna (1976) presented findings that suggest that high-achieving students learn more efficiently if permitted to proceed at their own pace. Verification feedback provides the level of information most helpful in this endeavor.
30	For low-achieving learners, use correct response and some kind of elaboration feedback.	Using the same rationale as with supplying scaffolding to low-achieving students, the prescription here is to ensure low-achieving students receive a concrete, directive form of feedback support (e.g., Clariana, 1990; Hanna, 1976).
31	For learners with low learning orientation (or high performance orientation), give specific feedback.	As described in the study by Davis et al., (2005), if students are oriented more toward performance (trying to please others) and less toward learning (trying to achieve an academic goal), provide feedback that is specific and goal directed. Also, keep the learner's eye on the learning goal (Hoska, 1993).

C Pruebas de programación y sistemas de información

C.1 Pruebas 1 programación



PRUEBA 0 PROGRAMACION

NOMBRE: _____

CURSO: PROGRAMACIÓN SEC.2 STGO S-SEM. 2017/1

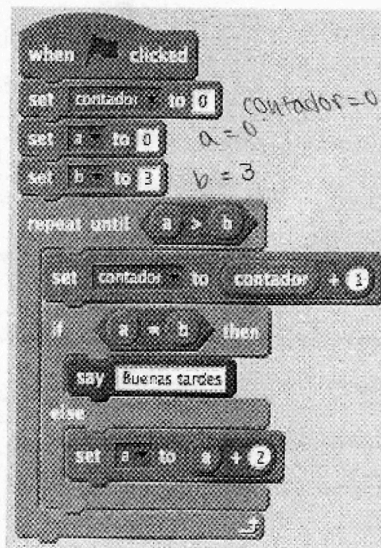
PÁGINA: 1 DE 16



TIEMPO: 99 minutos (1 minuto por punto)

LENGUAJE SCRATCH (TOTAL: 10 puntos)

1. a) (3 puntos) Para el siguiente código, completa la tabla con los valores que las variables *a*, *b* y *contador* van tomando cada vez que cualquiera de ellas cambia de valor. Las 3 primeras líneas ya están completas. Completa el resto.



contador	a	b
0	No existe	No existe
0	0	No existe
0	0	3
1	0	3
1		3
2	2	3
2	4	3

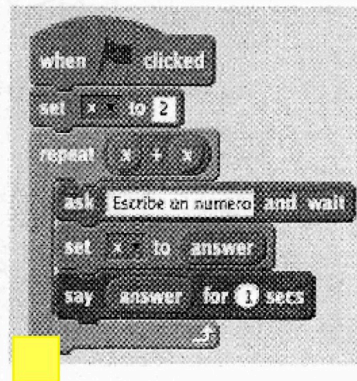
- b) (2 puntos) Escribe lo que este programa despliega por pantalla y justifica.

la pantalla no va a desplegar Buenas tardes ya que a nunca llega ser igual a b, por lo cual solo despliega el término del programa.





2. (5 puntos) Escribe en lenguaje C el equivalente funcional del siguiente programa en Scratch. Nota: el equivalente funcional en C a esperar 1 segundo en Scratch es `sleep(1)`. La función `sleep()` se encuentra definida en la biblioteca `unistd.h`.



```
#include <cs50.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

int main(void) {
    int x = 2;
    for (int i = 0; i < x + x; i++) {
        printf("Escribe un numero ");
        sleep(1);
        answer = GetInt();
        int x = answer;
        printf("i% ", answer);
    }
}
```





INTRODUCCIÓN AL LENGUAJE C (TOTAL: 48 puntos)

3. (8 puntos) En la mayoría de los juegos de video, los personajes tienen que sortear distintos obstáculos, como por ejemplo, una escalera. Escribe un programa en C que solicite al usuario ingresar el número de escalones de una escalera por la que se sube de izquierda a derecha y despliegue en pantalla la escalera correspondiente con *hashtags*. Por ejemplo, si el usuario ingresa el número 4, el programa desplegaría lo siguiente:

```
#
##
###
####
```

```
#include <stdio.h>
#include <cs50.h>
#include <string.h>

int main(void) {
    int x;
    printf("ingresa el numero de escalones");
    x = GetInt();
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < x - i; j++) {
            printf("#");
        }
        printf("\n");
    }
}
```





4. (8 puntos) Escribe el código de un programa que solicite al usuario ingresar un número positivo múltiplo de 2 y 3 y que verifique que efectivamente el usuario ingresó un número que sea positivo, múltiplo de 2 y múltiplo de 3. Si el usuario ingresa un número que no cumple esta condición, el programa debe indicarlo y solicitar un nuevo número.

```
#include <string.h>
#include <cs50.h>
#include <stdio.h>

int main(void) {
    int x = GetInt();
    while ((x % 2 != 0) && (x % 3 != 0) && (x > 0)) {
        printf("ingresa un numero");
    }
}
```

5. (3 puntos por cada error) Un alumno de programación implementó un programa para imprimir en pantalla los números pares contenidos en un arreglo de enteros. El alumno escribió el siguiente código:

```
1. void main()
2. {
3.     int numeros[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
4.     int N = 100;
5.     for (int i = 1; i <= N; i++)
6.     {
7.         if (i % 2 == 0)
8.             printf("%d\n", numeros[i]);
9.     }
10. }
```





Pero este código tiene al menos 3 errores de compilación y 3 de ejecución. Completa la siguiente tabla donde debes identificar un error de compilación y un error de ejecución, indicando cómo solucionarlo.

Error de compilación	
Descripción del error:	coloca %s que se refiere a un string siendo que no se utiliza
Solución:	cambiar %s por %i
Error de ejecución	
Descripción del error:	no evalúa la posición 0 en el arreglo
Solución:	cambiar $int + i = 1$ por $int + i = 0$.





6. (3 puntos por cada error) El siguiente código tiene 2 errores de ejecución. El objetivo del programa es imprimir en orden alfabético (comparando sólo la primera letra) los 2 nombres ingresados por el usuario. Por ejemplo, si el usuario ingresa los nombres Juan y Ana, por pantalla debe aparecer:

Ana
Juan

```

1. #include <stdio.h>
2. #include <cs50.h>
3.
4. int main(void)
5. {
6.     string nombres[2];
7.
8.     for(int i =0; i<2; i++)
9.     {
10.         printf("Ingresa nombre:\n");
11.         nombres[i]=GetString();
12.     }
13.
14. if(nombres[0][0]>nombres[1][0])
15.     printf("%s\n%s\n", nombres[0], nombres[1]);
16. else
17.     printf("%s\n%s\n", nombres[0], nombres[1]);
18. }
```

Completa la siguiente tabla, indicando la línea en la que se produce el error, describiéndolo y explicando cómo solucionarlo.

Línea en la que se produce el primer error:	
Descripción:	va a guardar el nombre en el mismo casillero.
Solución	en vez de nombres[0] colocar nombres[i]





Línea en la que se produce el segundo error:	
Descripción	compara los strings como números sin hacer type casting.
Solución	en vez de if (nombres[0][20] > nombres[1][20]) colocar if ((int) nombres[0][20] > (int) nombres[1][20])

7. (10 puntos) Un estudiante comenzó a escribir el programa que permite verificar si una palabra ingresada por el usuario es palíndromo. Una palabra palíndromo es aquella que se lee igual de izquierda a derecha, que de derecha a izquierda. Por ejemplo: reconocer, oso. La palabra *cuadrado* en cambio no es palíndromo porque de derecha a izquierda se lee *odardauc*. Lamentablemente, el estudiante tuvo que hacer otras cosas y no alcanzó a escribir el código de la función `palindrome()`. Esta función recibe una variable string como argumento y retorna 1 (true) si es que la palabra es palíndromo y 0 (false) en caso contrario. Escribe el código de la función `palindrome` que permita la correcta ejecución del programa.

```
//Incluye cualquier biblioteca que haga falta
#include <stdio.h>
#include <cs50.h>

//Define el prototipo de la función palindrome() aquí

int main()
{
    string palabra;

    printf("Ingrese una palabra: ");
    palabra = GetString();

    if (palindrome(palabra))
        printf("Palabra palíndromo");
}
```





```
else
    printf("Tu palabra NO es palíndromo");
return 0;
```

// describe el código de la función palindrome aquí

```
void palindrome( string palabra ) {
```

```
    arr [strlen(palabra);
```

```
    for (int i=0; i < strlen(palabra); i++) {
```

```
        arr [strlen(palabra)
```

```
arr [palabra] =
```

8. (10 puntos) Wikipedia define una Serie de Taylor como una aproximación de funciones mediante una suma de potencias enteras de polinomios, llamados términos de la serie.

En este problema deberás implementar una función en C llamada `geometrica()` que aproximará el valor de la expresión $a/(1-x)$ usando su serie de Taylor:

$$\frac{a}{(1-x)} = \sum_{i=0}^{\infty} a \cdot x^i$$





La función deberá recibir 3 parámetros de entrada: el valor de a , el valor de x y el número de términos de la sumatoria (ya que en el computador no hacemos ciclos infinitos) y retornar el número resultante de la sumatoria. Por ejemplo, `geometrica(10, 0.5, 100)` debe calcular la siguiente sumatoria:

$$\sum_{i=0}^{99} 10 \cdot 0.5^i = 10 \cdot 0.5^0 + 10 \cdot 0.5^1 + 10 \cdot 0.5^2 + \dots + 10 \cdot 0.5^{99}$$

y retornar su resultado.

Escribe un programa en C que:

- solicite al usuario los 3 valores necesarios (la constante a , el valor de x y el límite superior de la sumatoria)
- despliegue por pantalla el resultado obtenido al llamar a la función `geometrica()` con los valores ingresados por el usuario
- implemente la función `geometrica()` recién descrita

HINT: la función `pow(a,b)` de la librería `math.h` retorna el resultado a^b . Por ejemplo, `pow(2,3)` retorna el double 8.0

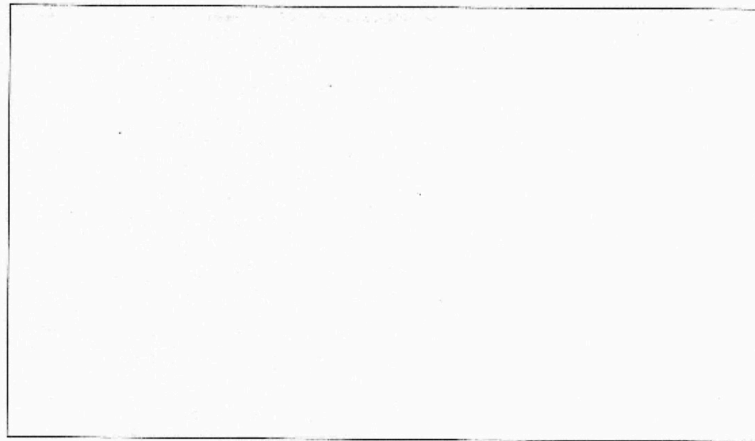
```
#include <stdio.h>
#include <cs50.h>
#include <string.h>

double geometrica(int a, float x, int limite);

int main(void) {
    int a = GetInt();
    float x = GetFloat();
    int limite = GetInt();
    printf("Resultado: %f\n", geometrica(a, x, limite));
}

double geometrica(int a, float x, int limite) {
    double total = 0;
    for (int i = 0; i < limite; i++) {
        total += a * pow(x, i);
    }
    return total;
}
```



**BÚSQUEDA BINARIA Y ORDENAMIENTO (TOTAL: 41 puntos)**

9. a) (4 puntos) Tienes un arreglo de 10^3 números desordenados en el que debes buscar un determinado número por **una única vez**. En términos del número de comparaciones, ¿cuál de las siguientes alternativas es mejor? **Alternativa a)** aplicar búsqueda lineal. **Alternativa b)** aplicar selection sort y luego búsqueda binaria. Justifica tu elección.

alternativa a ya que con la alternativa b buscaría el número más de una vez, ya que primero hay que ordenarlo lo cual lleva a buscar números y luego determinar el número lo cual llevaría a una segunda búsqueda.





- b) (4 puntos) ¿Hay algún caso en que resulte conveniente aplicar la alternativa no seleccionada en la pregunta anterior? Justifica.

se recomendaría si no importara la cantidad de veces que se buscara un número

10. El siguiente código implementa búsqueda binaria y no tiene errores.

```
1. #include <stdio.h>
2. #include <string.h>
3. #include <unistd.h> //para usar función sleep()
4. #include <cs50.h>
5.
6. int min=1; int max=100; int adivine=0;
7. string resp,max_mes;
8.
9. void busqueda_binaria(int min, int max);
10. int evaluacion(void);
11.
12. int main(void)
13. {
14.     int contador=0;
15.     printf("Piensa en un número entre 1 y 100\n");
16.     sleep(2);
17.     while(adivine!=0)
18.     {
19.         busqueda_binaria(min,max);
20.         adivine=evaluacion();
21.         contador++;
22.     }
23. }
```





```

24.
25. void busqueda_binaria(int min, int max)
26. {
27.     if(min==max) {
28.         printf("¿Pensaste en el número %i!\n", min);
29.         resp="sí"; }
30.     else {
31.         printf("¿Pensaste en el %i?(sí,no)\n", min-(max-
32. min)/2);
33.         resp=GetString(); }
34. }
35.
36. int evaluacion(void)
37. {
38.     if(strcmp(resp,"no")==0) {
39.         printf("¿mayor o menor? (mayor, menor)\n");
40.         may_men=GetString();
41.         if(strcmp(may_men,"mayor")==0)
42.             min=min+((max-min)/2)+1;
43.         else
44.             max=max-((max-min)/2)-1;
45.         return 0;
46.     }
47.     else {
48.         printf("Adiviné!\n");
49.         return 1;
50.     }
51. }
    
```

a) (10 puntos) Completa la siguiente tabla con los valores que toman las variables *contador*, *adivine*, *min* y *max* cada vez que una de ellas cambie de valor. Supón que el usuario piensa en el número 62 (y contesta correctamente las preguntas).

contador	adivine	min	max
0	0	1	100
1	0	80	100
2	0	50	75
3	0	50	65
4	0	50	65





5	0	32	64
6	1	1	

- b) (5 puntos) Escribe lo que se despliega en pantalla al ejecutar el código cuando el usuario piensa en el número 62 (supón que el usuario contesta correctamente todas las preguntas).

```

Pienso en un número entre 1 y 100.
¿Pensaste en el 30?
usuario = no
¿mayor o menor?
usuario = mayor
contador = 0
¿Pensaste en el 70?
usuario = no
¿mayor o menor?
usuario = mayor
contador = 1
¿Pensaste en el 85?
usuario = no
¿mayor o menor?
usuario = menor
contador = 2
    
```

11. (6 puntos) Para el arreglo de 7 números enteros que se muestra en la primera fila de la tabla siguiente, rellena las filas vacías para mostrar cómo va cambiando el contenido del





arreglo en la medida que se ejecuta el algoritmo **Insertion Sort**. Explica cómo fuiste llenando la tabla.

[43]	[5]	[13]	[19]	[5]	[18]	[23]
[5]	[43]	[13]	[19]	[5]	[18]	[23]
[5]	[13]	[43]	[19]	[5]	[18]	[23]
[5]	[13]	[19]	[43]	[5]	18	[23]
[5]	[5]	[13]	[19]	[43]	18	[23]
[5]	[5]	[13]	[18]	[19]	43	[23]
[5]	[5]	[13]	[18]	[19]	23	[43]

Explicación:

Insertion sort va a ir comparando el segundo número con el anterior, si este es mayor al primero no lo cambiará, de caso contrario lo cambia, al avanzar compara el siguiente número con los otros ya ordenados y lo colocará en su posición, donde este sea mayor al que está antes y menor al siguiente.





12. (4 puntos por cada función) Escribe el contenido de las funciones `swap()`, `busca_max()` y `selection_sort()`. La función `swap()` toma como datos de entrada dos enteros `a` y `b` e intercambia los contenidos del arreglo de esas posiciones (el número que estaba en la posición `a` queda en la posición `b` y viceversa). La función `busca_max(int inicio)` retorna la posición donde se encuentra el entero con mayor valor entre las posiciones `inicio` y final del arreglo. Finalmente, la función `selection_sort()` ordena los números de un arreglo de mayor a menor (es decir, el mayor número queda en la posición 0 del arreglo).

```
#include <stdio.h>
#include <stdlib.h>

int numeros[20];
int N=20;

void llena_arreglo(void);
void imprime_arreglo(void);
int busca_max(int inicio);
void selection_sort(void);
void swap (int a, int b);

int main(void)
{
    llena_arreglo();
    imprime_arreglo();
    selection_sort();
    imprime_arreglo();
}

void llena_arreglo(void)
{
    for(int i=0; i<20; i++)
        numeros[i] = rand()%20;
}

void imprime_arreglo(void)
{
    for(int i=0; i<N; i++)
        printf("[%i]: %i\n", i, numeros[i]);
}
```





```
void swap(int a, int b)
{
    int aux;
    for (int i = 0, < 20, i++) {
        if (numero[a] > numero[b]) {
            numero[a] = aux;
            numero[b] = numero[a];
            aux = numero[b];
        }
    }
}
```

```
int busca_max(int inicio)
{
    < >
}
```

```
void selection_sort(void)
{
    < >
}
```



C.2 Rúbrica pruebas 1 programación

P1.a	No se observa una lógica en los números ingresados (aun cuando haya llegado al resultado correcto).	Presenta 6 errores (se descuenta 0.25 por error)	Completa todo el cuadro correctamente	
	0puntos	1.5puntos	3puntos	
P1.b	No contesta o lo que contesta está errado	Señala que no hay despliegue, pero no indica la razón	Señala que no hay despliegue y explica la razón (condición a=b nunca se cumple)	
	0puntos	1puntos	2puntos	
P2.a	no incluye libreria ni inicializa variable x	incluye libreria XOR inicializa variable x	incluye libreria e inicializa variable x	
	0puntos	0.5puntos	1puntos	
P2.b	No implementa el ciclo o lo hace con errores graves	Implementa el ciclo con errores menores	Implementa el ciclo correctamente (inicio, condición, actualización)	
	0puntos	1puntos	2puntos	
P2.c	No solicita dato al usuario ni despliega datos	Solicita datos al usuario XOR los despliega con errores menores	Solicita datos al usuario y los despliega correctamente	
	0puntos	1puntos	2puntos	
P3.a	no incluye libreria, ni declara variables, ni solicita lo requerido	Incluye correctamente bibliotecas XOR declara correctamente las variables utilizadas en el programa AND Presenta errores menores en la impresión o solicitud de datos	Declara en forma correcta todas las variables utilizadas en el programa e incluye las bibliotecas necesarias AND Uso correcto de funciones printf y GetInt	
	0puntos	1puntos	2puntos	
P3.b	No usa ciclos anidados o lo hace con errores graves	Usa ciclos anidados, pero con errores en los límites	Usa ciclos anidados correctamente	
	0puntos	2puntos	4puntos	
P3.c	No despliega los escalones o lo hace con errores mayores	Despliega los escalones con errores menores (e.e. Falta salto de línea, símbolos intercambiados)	Despliega correctamente los escalones	
	0puntos	1puntos	2puntos	
P4.a	no incluye libreria ni declara variables	incluye libreria XOR declara variables	incluye libreria y declara variables	
	0puntos	1puntos	2puntos	
P4.b	No verifica que el número ingresado por el usuario sea positivo y múltiplo de 2 y 3	Verifica que el número ingresado por el usuario sea positivo o múltiplo de 2 y 3, pero no ambas	Verifica que el número ingresado por el usuario sea positivo y múltiplo de 2 y 3, pero con errores menores	Verifica que el número ingresado por el usuario sea positivo y múltiplo de 2 y 3
	0puntos	1puntos	2puntos	4puntos
P4.c	No vuelve a solicitar el número al usuario	Solicita en forma correcta el re-ingreso del número		
	0puntos	2puntos		

P5.a	No describe ningún error en forma correcta	Describe un error pero presenta errores o poca clarificación del error. Ejemplos: El segundo nombre se borra. o accede dos veces a posición 0	Describe ambos errores pero presenta errores o poca clarificación del error	Describe un error en forma precisa	Describe un error en forma precisa y en el otro presenta errores o poca clarificación del error	Describe ambos errores en forma precisa
	0puntos	0.5puntos	1puntos	1.5puntos	2puntos	3puntos
P5.b	No describe ninguna solución en forma correcta	Escribe una nueva línea de código pero presenta al menos un error	Escribe dos nuevas líneas de código pero presenta al menos un error en cada una	Escribe una solución en forma precisa	Escribe una solución en forma precisa. y en la otra presenta al	Escribe las dos soluciones en forma precisa
	0puntos	0.5puntos	1puntos	1.5puntos	2puntos	3puntos
P6.a	No declara ninguna línea de error	Declara en forma correcta una línea de error	Declara en forma correcta dos líneas con error			
	0puntos	0.5puntos	1puntos			
P6.b	No describe ningún error	Describe un error pero presenta errores o poca clarificación del error. Ejemplos: El segundo nombre se borra. o accede dos veces a posición 0	Describe dos errores pero presenta errores o poca clarificación del error	Describe un error en forma precisa	Describe un error en forma precisa y en el otro presenta errores o poca clarificación del error	Describe ambos errores en forma precisa
	0puntos	0.5puntos	1puntos	1.25puntos	1.75puntos	2.5puntos
P6.c	No describe ninguna solución en forma correcta	Escribe una nueva línea de código pero presenta al menos un error	Escribe dos nuevas líneas de código pero presenta al menos un error en cada una	Escribe una solución en forma precisa	Escribe una solución en forma precisa. y en la otra presenta al	Escribe las dos soluciones en forma precisa
	0puntos	0.5puntos	1puntos	1.25puntos	1.75puntos	2.5puntos
P7.a	No crea una función	Crea una función pero de manera incorrecta (faltan parámetros o tipo de retorno incorrecto)	Declara de forma correcta la función			
	0puntos	1puntos	2puntos			
P7.b	No usa ciclo para recorrer palabra	Usa ciclo. pero comete errores menores	Usa ciclo correctamente			
	0puntos	1puntos	2puntos			
P7.c	No verifica condición palíndromo	Verifica con errores menores (e.g. Error en índice de arreglo)	Verifica correctamente la condición de palíndromo			
	0puntos	2puntos	4puntos			
P7.d	No retorna valor o lo hace con errores graves	Retorna con errores menores	Retorna correctamente el resultado de la función			
	0puntos	1puntos	2puntos			
P8.a	No crea una función	Crea una función pero de manera incorrecta (faltan parámetros o tipo de retorno incorrecto)	Declara de forma correcta la función			
	0puntos	1puntos	2puntos			
P8.b	No llama a la función creada	Llama a la función creada pero no almacena o muestra el resultado	Llama a la función y almacena o muestra el resultado en forma correcta			

P8.c	No usa ciclos for/while o los usa de mala manera. ni calcula en forma adecuada la suma	Usa un ciclo for o while. pero tiene la inicialización o condiciones erróneas	Usa una variable para acumular el valor de la suma. pero no usa ciclo for	Usa una variable para acumular el valor de la suma. y el ciclo for/while tiene la inicialización o condiciones erróneas	Usa la suma y el ciclo for/while en forma correcta		
	0puntos	0.3puntos	1puntos	1.3puntos	2puntos		
P8.d	No usa return para retornar el valor calculado	Usa return para devolver un dato. pero no es el dato correcto	Usa return en forma correcta				
	0puntos	0.3puntos	1puntos				
P8.e	No pide los datos al usuario	Pide datos al usuario pero no los almacena o usa en forma correcta	Pide datos al usuario. pero dentro de la función creada y no de main	Pide datos al usuario de forma correcta			
	0puntos	0.3puntos	0.7puntos	1puntos			
P8.f	No usa printf de forma correcta	Usa printf para imprimir mensajes en pantalla. pero no utiliza % para intercalar valores	Usa printf de forma correcta				
	0puntos	0.3puntos	1puntos				
P8.g	No declara variables ni librerías	Declara en forma correcta las librerías O las variables utilizadas en el programa	Declara en forma correcta todas las variables utilizadas en el programa y las librerías				
	0puntos	0.5puntos	1puntos				
P9.a	No responde. responde incorrectamente. o no justifica selección	Selecciona la alternativa correcta XOR Explica correctamente la situación justificada	Selecciona la alternativa correcta AND Explica correctamente la situación justificada				
	0puntos	2puntos	4puntos				
P9.b	No identifica ni explica correctamente situación solicitada	Identificada XOR explica correctamente situación solicitada	Identificada AND explica correctamente situación solicitada				
	0puntos	2puntos	4puntos				
P10.a	no llena la tabla o esta totalmente errónea	Contiene 19 errores (descontar 0.4 por error)	Contiene 12 errores (descontar 0.4 por error)	Contiene 6 errores (descontar 0.4 por error)	Rellena toda la tabla en forma correcta (descontar 0.4)		
	0puntos	2.4puntos	5.2puntos	7.4puntos	10puntos		
P10.b	No escribe nada	Escribe alguna de las preguntas con omisiones menores	Escribe piensa	Escribe piensa y 2 preguntas por omisiones menores	Escribe piensa y 3 preguntas por omisiones menores	Escribe Piensa. y las dos primeras preguntas en forma correcta	Escribe Piensa. y las tres preguntas en forma correcta
	0puntos	0.7puntos	1puntos	2.4puntos	3.1puntos	4puntos	5puntos
P11.a	No escribe ninguna fila o lo hace con errores graves	Identifica solo algunos cambios. escribe al menos la mitad de las filas	Identifica correctamente 9 instantes en que cambia el contenido del arreglo				
	0puntos	1.5puntos	3puntos				

P11.b	Escrebe el contenido del arreglo con errores graves	Escrebe correctamente el contenido del arreglos en algunos de los cambios identificados	Escrebe correctamente el contenido del arreglo en cada cambio identificado	
	<i>0puntos</i>	<i>1.5puntos</i>	<i>3puntos</i>	
P12.a	No responde o lo que responde no tiene sentido	Utiliza variable auxiliar, pero se equivoca en el orden de las instrucciones, borrando uno de los datos	Swap correctamente implementado	
	<i>0puntos</i>	<i>2puntos</i>	<i>4puntos</i>	
P12.b	No responde o lo que responde no tiene sentido	Recorre el arreglo, pero no encuentra correctamente el índice del máximo elemento	Encuentra el índice del máximo elemento	
	<i>0puntos</i>	<i>2puntos</i>	<i>4puntos</i>	
P12.c	No responde o lo que responde no tiene sentido	Comete 2 errores en los siguientes aspectos: recorrido del arreglo. llamar a busca_max() y luego swap()	Comete 1 error de los siguientes aspectos: recorrido del arreglo. llamar a busca_max() y luego swap()	Recorre arreglo correctamente. busca máximo y realiza swap
	<i>0puntos</i>	<i>1puntos</i>	<i>2puntos</i>	<i>4puntos</i>

C.3 Pruebas 2 programación



PRUEBA 1 PROGRAMACIÓN
NOMBRE [REDACTED]
CURSO: PROGRAMACIÓN SEC.2 STGO S-SEM. 2017/1
PÁGINA: 1 DE 8



PROGRAMACIÓN PRUEBA 1

1. **[1 punto]** Para el código que se muestra dentro del recuadro, implementa la **función recursiva producto_punto(a,b,n)** que retorna el producto punto (entero) entre dos vectores a y b, de largo n. El producto punto entre dos vectores a y b, se define como:

$$a \cdot b = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

```
#include <stdio.h>
#include <cs50.h>
#include <time.h>

//escribe el prototipo de la función aquí
void producto_punto(a[], b[], int c);

int main(void){
    int a[10], b[10];
    srand(time(NULL));
    llena(a);
    llena(b);
    printf("El producto punto de a y b es: %i\n",
    producto_punto(a,b,10));
}

void llena(int a[]) {
    for(int i=0; i<10; i++)
        a[i]=rand()%5 + 1;
}

//escribe el código de la función recursiva aquí
void producto_punto(a[], b[], int c){
    if (c == 0)
        return 0;
    else
        if (c != 0) {
            a[0] * b[0];
            return producto_punto + producto_punto(a[0]+1, b[0]+1, c-1);
        }
}
```





2. (0.5 puntos) El siguiente es el pseudo-código del algoritmo Merge Sort (N se refiere al tamaño del arreglo a ordenar):

```
merge_sort(arreglo)
{
    if N == 1
        return;
    else
    {
        arreglo_izq=mitad izquierda del arreglo;
        arreglo_der=mitad derecha del arreglo;
        merge_sort(arreglo_izq);
        merge_sort(arreglo_der);
        merge(arreglo_izq, arreglo_der);
    }
}
```

Para el arreglo {73,48,37,21,13}, lista el orden en que se ejecutan las funciones y los datos con los que operan.

merge {73, 48, 37, 21, 13}
 merge izquierda {73, 48, 37}
 merge izquierda {73}
 merge derecha {48}
 merge {48, 73}
 merge derecha {37}
 merge {37, 48, 73}
 merge derecha {21, 13}
 merge izquierda {21}
 merge derecha {13}
 merge {13, 21}
 merge {13, 21, 37, 48, 73} (final)





3. (3.5 puntos) Ud. es dueño de 15 ferias donde se vende todo tipo de verduras y frutas. Cada feria tiene varios locales y de cada local se tiene la siguiente información: nombre y apellido del dueño, rubro (verduras o frutas), número del local (un número entre 0 y 29), el año de ingreso a la feria y el código de la feria a la que pertenece (un número entre 0 y 14). La información de todos los locales se almacena en el archivo "locales.txt". Por ejemplo, una línea del archivo "locales.txt" sería:

Juan Leiva Frutas 5 1997 14

Por su parte, la información de todas las ferias se almacena en el archivo "ferias.txt". Cada una de las 15 líneas de este archivo almacena el código de la feria (un número entre 0 y 14), el nombre de la feria y el número de locales activos (una feria tiene espacio para un máximo de 30 locales). Un ejemplo de este archivo es:

12 Valparaíso 30
14 Vega_Central 17
...
1 Viña_del_Mar 24

Para administrar las ferias, se ha creado un programa con la siguiente estructura general:

```
#include <stdio.h>
#include <cs50.h>

void lee_feria(void);
void lee_locales(void);
float años_prom(struct feria f);
void ingresa_local(void);
void escribir_locales(void);

struct local{
    string nombre;
    string apellido;
    string rubro;
    int año_ing;
} int cod_loca;
```





```

struct feria{
    string nombre;
    struct local locales[30];
    int num_locales;
} int cod_feria;

struct feria f[15];

int main (void)
{
    lee_feria(); //suponga que existe
    lee_locales(); //suponga que existe
    printf("El promedio de años de los locales de la feria
    Vega Central es: %f\n", años_prom(f[14]));
    ingresa_local();
    escribir_locales();
}
    
```

Las funciones lee_feria() y lee_locales() ya existen. La primera abre el archivo ferias.txt y almacena su contenido en el arreglo de tipo struct feria llamado f. La segunda abre el archivo locales.txt y almacena su contenido en la estructura correspondiente.

Se le pide que complemente este programa con las funcionalidades que se piden a continuación (no es necesario declarar los prototipos de las funciones antes de main()).

a) (1 punto) Cree la función años_prom() la que retorna el promedio del número de años que tienen los locales de una feria determinada. La función recibe como argumento de entrada una variable de tipo struct feria.

```

float años_prom(struct feria f){
    float suma = 0;
    int contador = 0;
    for (int i = 0; i < struct feria f; i++) {
        suma = suma + f[i].años_ing;
        contador++;
    }
    return suma / (float) contador;
}
    
```





b) (1 punto) Cree una función `ingresa_local()` que permite ingresar un local nuevo en la estructura correspondiente (Nota: esta función **no** escribe en el archivo). La función solicita al usuario: nombre y apellido del dueño del local, rubro, año de ingreso y código de la feria. El número del local no se pregunta al usuario, sino que se asigna como tal el número de locales activos en la feria. Note que este número debe ser actualizado donde corresponde. Verifique si la feria está llena, en cuyo caso de debe desplegar por pantalla el mensaje "No es posible crear un nuevo local. Feria llena".

```
void ingresa_local() {
    int c = 0;
    while (c == 30) {
        for (int a = 0; a < 15; a++) {
            for (int i = 0; i < 30; i++) {
                printf("ingrese nombre: ");
                f[a].locales[i].nombre = GetString();
                printf("ingrese apellido ");
                f[a].locales[i].apellido = GetString();
                printf("ingrese el rubro: ");
                f[a].locales[i].rubro = GetString();
                printf("ingrese el año de ingreso: ");
                f[a].locales[i].anio_ing = GetInt();
                printf("ingrese código feria ");
                f[a].cod_feria = GetInt();
                f[a].locales[i].cod_local = c;
                c++;
            }
        }
        printf("No es posible crear un nuevo local. Feria llena");
    }
}
```





- c) **(1 punto)** Crea la función `escribir_locales()` que sobrescribe el archivo `locales.txt` con los datos de los locales.

```
void escribir_locales() {
    FILE * arch;
    arch = fopen("locales.txt", "w");
    for (int a = 0; a < 15; a++) {
        for (int i = 0; i < 30; i++) {
            fprintf(arch, "%s %s %i %i %i\n",
                f[a].locales[i].nombre, f[a].locales[i].apellido,
                f[a].locales[i].cod-local, f[a].locales[i].anio-ing,
                f[a].cod-feria);
        }
    }
}
```

- d) **(0.5 puntos)** El Ministerio de Salud hizo obligatorio que cada local tenga una resolución sanitaria, esto es un número identificador que autoriza al local a operar. Haga los cambios necesarios en las estructuras para que se pueda almacenar este código.

```
struct local {
    string nombre;
    string apellido;
    string rubro;
    int anio-ing;
    int cod-local;
    int resolucion-sanitaria;
}
```



**PREGUNTA 4 (1 puntos)**

- a) (0.5 puntos) Escribe lo que se despliega en pantalla al ejecutar el siguiente código.

```
#include <stdio.h>

int main()
{
    int *a;
    int *b;
    int x=3;

    b=&x;?
    a=&x;?
    *a=0;

    printf(" *a = %d\n", *a);
    printf(" *b = %d\n", *b);
    printf(" x = %d\n", x);

}
```

Despliegue en pantalla:

```
*a = 0
*b = 0;
x = 3;
```





- b) (0.5 puntos) Escribe el código de la función `magic()` que recibe valores por referencia de las variables `a`, `b` y `x`, para que la función reemplace `x` por la suma de `a` y `b` (esto es `x=a+b`) e indique qué se muestra en pantalla.

```
#include <stdio.h>

//escribe el prototipo de la función aquí
void magic(int* a, int* b, int* c);

int main() {
    int *a, *b, x=3;

    b=&x;
    a=&x;
    magic(a,b,&x);
    printf("%d\n",x);
}

//escribe la función aquí
void magic(int* a, int* b, int* c) {
    *c = *a + *b;
}
```

Despliegue en pantalla:

6



C.4 Rúbrica pruebas 2 programación

P1	no escribe nada, o escribe una función simple no-recursiva	Identifica la condición de término de la recursión.	Identifica el caso general de recursión pero no la condición de término	Escribe bien la función recursiva con el caso base y caso general	
	0puntos	0.3puntos	0.7puntos	1puntos	
P2a	No responde o lo hace con errores graves de comprensión de recursión	Comete algunos errores (ya sea de orden, de nombre de funciones o de argumentos)	Escribe correctamente y en orden todos los llamados a las funciones merge_sort y merge, con los argumentos correctos		
	0puntos	0.25puntos	0.5puntos		
P3a	No se declara la función ni sus parámetros	Se declara la función pero de forma incorrecta o bien con los parámetros de forma incorrecta	Declara bien la función, pero no utiliza ciclos para el cálculo o bien no asigna bien los valores a calcular	Declara bien la función y utiliza ciclos pero tiene errores menores en el cálculo, o bien no retorna la variable correcta	Declara bien la función, calcula de forma correcta y retorna el valor que corresponde
	0puntos	0.2puntos	0.5puntos	0.8puntos	1puntos
P3b	No declara la función	Se declara la función pero de forma incorrecta o bien con los parámetros de forma incorrecta	Declara bien la función pero no solicita al usuario todos los valores necesarios o bien solicita al usuario correctamente los valores pero no asigna el valor del código dependiendo de los números libres que haya	Asigna bien los valores del código pero no verifica que la feria esté llena o bien verifica que la feria esté llena pero no muestra el mensaje en pantalla	Declara bien la función, solicita los valores y asigna de forma correcta el código mandando un mensaje de alerta si la feria está llena
	0puntos	0.2puntos	0.5puntos	0.8puntos	1puntos
P3c	No declara la función	Crea la función, abre el archivo de forma correcta pero no hace las operaciones de forma correcta para escribir los datos	Crea la función, abre el archivo de forma correcta, utiliza ciclos para asignar la información al archivo pero comete errores al escribir en archivo	Crea la función, abre el archivo de forma correcta, utiliza ciclos para asignar la información al archivo, escribe correctamente la información en archivo pero no cierra el archivo	Crea la función, abre el archivo de forma correcta, utiliza ciclos para asignar la información al archivo, escribe correctamente la información en archivo y cierra el archivo de forma correcta
	0puntos	0.2puntos	0.6puntos	0.8puntos	1puntos
P3d	No asigna la variable en ninguna parte	Asigna la variable pero fuera de la estructura o de forma incorrecta	Asigna la variable a la estructura local de forma correcta		
	0puntos	0.2puntos	0.5puntos		
P4a	No responde o responde algo incoherente	Responde <code>"*a = 0"</code> o <code>"*b=0"</code> o <code>"x=0"</code>	Responde <code>("*a = 0" y "*b=0")</code> o <code>("*a = 0" y x=0")</code> o <code>("*b=0" y "x=0")</code>	Responde <code>"*a = 0" y "*b=0" y "x=0"</code>	
	0puntos	0.15puntos	0.3puntos	0.5puntos	
P4b	No responde o responde algo incoherente o dentro de la función magic indica <code>x=a+b</code>	Escribe dentro de la función magic <code>*k=*i+j</code> o dice que el printf imprime "Muestra un 6"	Escribe dentro de la función magic <code>*k=*i+j</code> y dice que el printf imprime "Muestra un 6"		
	0puntos	0.25puntos	0.5puntos		

C.5 Examen programación



EXAMEN
NOMBRE: [REDACTED]
CURSO: PROGRAMACIÓN SEC.2 STGO S-SEM. 2017/1
PÁGINA: 1 DE 6



PROGRAMACIÓN EXAMEN

1. (1.2 puntos) La estructura de la izquierda guarda, para cada alumno, los datos personales y las notas de las pruebas P0 y P1 (los alumnos que no han rendido una prueba tienen un 1.0). La estructura de la derecha, guarda las notas de la prueba acumulativa de CS50.

<pre>struct notasPruebas { string nombre; string apellido; string RUT; float notas[2]; };</pre>	<pre>struct notasAcum { string nombre; string apellido; string RUT; float nota; };</pre>
---	--

De los 600 alumnos de CS50, solo 200 rindieron la prueba acumulativa. En el siguiente código se usan dos arreglos para almacenar las notas:

- en el arreglo CS50[600] se guardan las notas de las pruebas P0 y P1
- en el arreglo acumulativa[200] se almacenan las notas de la prueba acumulativa

Completa el código de la función main() para que el programa reemplace (en el arreglo CS50) la peor nota de las pruebas P0 y P1 por la nota de la prueba acumulativa. El reemplazo ocurre solo en caso que la nota de la prueba acumulativa sea mayor a alguna de las notas de la prueba. Atención: el alumno de CS50[i] no necesariamente corresponde al alumno de acumulativa[i].

```
#include <stdio.h>  
#include <string.h>  
  
int main(){  
    struct notasPruebas CS50[600];  
    struct notasAcum acumulativa[200];  
    leerNotasCS50(CS50); //Supón que esta función existe y funciona  
    leerNotasAcum(acumulativa); //Supón que esta función existe y funciona
```

//Escribe tu código al reverso

~~XXXXXXXXXXXXXXXXXXXX~~

~~XXXXXXXXXXXXXXXXXXXX~~

// escribí en el reverso



2. (1.2 puntos) En el archivo patologías.txt se almacenan los datos de los pacientes de un consultorio. Cada línea corresponde a los datos de un paciente: RUT y condición médica, separados por un espacio tal como se muestra a continuación:

8.234.324-k diabetes
9.345.654-2 hipertensión
7.482.477-8 diabetes

Implementa la función `int frecuencia(string enfermedad)` que retorne el número de pacientes que tienen `enfermedad`.

```
#include <stdio.h>
#include <string.h>
#include <cs50.h>

int main(){
    int f;
    string pat;

    printf("Ingrese patologia: ");
    pat=GetString();
    f=frecuencia(pat);
    printf("La frecuencia de la patologia '%s' es %i\n",pat,f);
}

//implementa tu función aqui
int frecuencia ( string enfermedad) {
    int s = 0;
    FILE * archivo;
    archivo = fopen ("patologías.txt", "r");
    string arr [500];
    while (feof != 1){
        fscanf ("patologías.txt", "%s", arr [500]);
    }
    for (int i=0; i<500; i++) {
        if (i % 2 == 0) {
            if (!strcmp (enfermedad, arr[i])) {
                s = s + 1;
            }
            else {
                return 0;
            }
        }
        else {
            return 0;
        }
    }
    return s;
}
```



3. (1.2 puntos) Las flechas numeradas identifican instantes de tiempo de ejecución del siguiente código. Por ejemplo: 1-> señala el instante inmediatamente después de ejecutar la instrucción `*b=&a`, y como ejemplo se muestra los valores en la tabla. Completa la tabla con los valores de las variables en los instantes de tiempo señalados. Si la variable no existe, o no puede ser modificada, marque una X en el casillero.

```
#include <stdio.h>
#include <stdlib.h>

void cambios(int x, int *y);

int main(void)
{
    int a=10;
    int *b=&a;
1->
    cambios(a, b);
4->
}

void cambios(int x, int *y)
{
    x++;
    y=&x;
2->
    *y=x+*y;
3->
}
```

	*b	x	*y
1->	10	X	X
2->	11	11	
3->	22	11	22
4->	22	X	X



4. (1.2 puntos) Implementa el código de la función `ordena_arreglo()` que recibe como argumento de entrada el nombre del arreglo letras minúsculas y lo deja ordenado alfabéticamente. Recuerda que el código ASCII de las letras minúsculas va desde el 97 (a) hasta el 122 (z).

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int N=15;

void llena_arreglo(char arreglo[]);
void ordena_arreglo(char arreglo[]);
void imprime_arreglo(char arreglo[]);

int main(void)
{
    char letras[N];
    srand(time(NULL));
    llena_arreglo(letras);
    imprime_arreglo(letras);
    ordena_arreglo(letras);
    imprime_arreglo(letras);
}

void llena_arreglo(char arreglo[])
{
    for(int i=0; i<N; i++)
        arreglo[i]=rand()%26+97;
}

void imprime_arreglo(char arreglo[])
{
    for(int i=0; i<N; i++)
        printf("%c ", arreglo[i]);
    printf("\n");
}

//escribe tu código al reverso de esta hoja
```




5. (1.2 puntos) Escribe el código de la **función recursiva** `pinta()` que recibe como argumento de entrada un número entero e imprime en pantalla ese número de símbolos `#`. La línea termina con un salto de línea. Por ejemplo:

Ingresa el número de símbolos a imprimir:

5

#####

```
#include <stdio.h>
#include <cs50.h>

void pinta(int n);

int main()
{
    printf("Ingresa el número de símbolos a imprimir:\n");
    int n=GetInt();
    pinta(n);
}

//escribe tu código aquí
```

```
void pinta (int n) {
    if (n = 1) {
        printf("#");
    }
    else {
        printf("#");
        pinta (n-1);
    }
    printf("\n");
}
```



EXAMEN
NOMBRE: [REDACTED]
CURSO: PROGRAMACIÓN SEC.2 STGC S-SEM. 2017/1
PÁGINA: 6 DE 6



C.6 Rúbrica examen programación

P1.a	No declara un ciclo para el arreglo "CS50" o lo hace con graves errores	Declara un ciclo para recorrer los 600 alumnos de CS50		
	0puntos	0.1puntos		
P1.b	No declara un ciclo para el arreglo "acumulativa" o lo hace con graves errores	Declara un ciclo para recorrer los 200 alumnos de la prueba "acumulativa"		
	0puntos	0.1puntos		
P1.c	No realiza una comparación de RUT entre ambos arreglos o lo hace con graves errores	Realiza una comparación, pero tiene errores menores en los índices o en los campos	Realiza la comparación sin errores empleando el campo RUT de ambos arreglos con STRCMP	
	0puntos	0.3puntos	0.5puntos	
P1.d	No realiza el reemplazo de notas correctamente	Realiza un reemplazo, pero lo hace con errores menores al no verificar la nota mayor	Realiza el reemplazo correctamente identificando la nota que debe reemplazarse	
	0puntos	0.2puntos	0.5puntos	
P2.a	No inicializa la variable que maneja la frecuencia	Declara e inicializa la variable que acumula la frecuencia		
	0puntos	0.2puntos		
P2.b	No abre el archivo, o lo hace con graves errores	Abre el archivo correctamente		
	0puntos	0.05puntos		
P2.c	No recorre el archivo y no lee los datos con la condición o lo hace con graves errores	Recorre el archivo, pero lee con errores menores los campos del archivo	Recorre el archivo, y lee los datos asignando correctamente los valores a las variables de condición	
	0puntos	0.2puntos	0.4puntos	
P2.d	No compara la condición buscada según el parámetro de la función o lo hace con graves errores	Compara la condición buscada dentro del ciclo, pero no acumula la variable de frecuencia	Compara y acumula correctamente la frecuencia según la enfermedad ingresada	
	0puntos	0.2puntos	0.4puntos	
P2.e	No cierra el archivo	Cierra el archivo correctamente		
	0puntos	0.05puntos		
P2.f	No retorna el resultado de la función	Retorna el resultado de la función		
	0puntos	0.1puntos		
P3. Línea 2	Todos los valores declarados en la fila son incorrectos	1 valor de la fila es correcto	2 valores de la fila son correctos	3 valores de la fila son correctos
	0puntos	0.1puntos	0.2puntos	0.3puntos
P3. Línea 3	Todos los valores declarados en la fila son incorrectos	1 valor de la fila es correcto	2 valores de la fila son correctos	3 valores de la fila son correctos
	0puntos	0.2puntos	0.4puntos	0.6puntos
P3. Línea 4	Todos los valores declarados en la fila son incorrectos	1 valor de la fila es correcto	2 valores de la fila son correctos	3 valores de la fila son correctos
	0puntos	0.1puntos	0.2puntos	0.3puntos

P4.A1	No declara un ciclo for para recorrer el arreglo	Declara un ciclo for para recorrer las N posiciones del arreglo	
	0puntos	0.2puntos	
P4.A3	No realiza una búsqueda del menor valor del arreglo	Implementa una función de búsqueda del menor valor, o lo hace dentro de la misma función con errores menores	Implementa una función o algoritmo adicional para buscar el menor valor del arreglo
	0puntos	0.3puntos	0.5puntos
P4.A4	No intercambia valores dentro del arreglo, o lo hace con graves errores	Realiza un intercambio, pero en posiciones incorrectas, como un valor con el siguiente u otro error.	Implementa una función que realiza el swap entre dos valores según el valor actual y valor mínimo del arreglo.
	0puntos	0.2puntos	0.5puntos
P5.a	No declara la función o lo hace con graves errores	Declara la función pero faltan parámetros	Declara la función correctamente
	0puntos	0.05puntos	0.1puntos
P5.b	No declara un caso base o término de la recursión	Declara un caso base, pero la instrucción de término tiene errores	Declara un caso base correcto y el último printf sin errores.
	0puntos	0.2puntos	0.4puntos
P5.c	No declara el caso recursivo (else) o lo hace con graves errores	Declara el caso recursivo, pero tiene errores menores en la instrucción printf	Declara correctamente el printf en el llamado recursivo
	0puntos	0.2puntos	0.4puntos
P5.d	No realiza el llamado a la función en forma recursiva	Realiza un llamado pero con parámetros incorrectos, o bien fuera del ELSE	Realiza un llamado a la función en forma recursiva correctamente
	0puntos	0.1puntos	0.3puntos

C.7 Pruebas 1 sistemas de información



PRUEBA 1 FORMA 1

NOMBRE: [REDACTED]

RUT: 17586810

CURSO: SISTEMAS DE INFORMACIÓN SEC.4 STGO FIC STGO 2017/1

PÁGINA: 1 DE 5



Prueba 1 Sistemas de Información

Los sistemas de car sharing, funcionan hace varios años en el mundo y Chile no es la excepción. En 2016 se conforma la compañía AWTO, que permite el arriendo de autos por cortos períodos de tiempo.

La flota de vehículos parte con 41 city cars y 5 autos híbridos los cuales están en las comunas de Providencia, Ñuñoa, Las Condes y La Reina. Para el segundo año de funcionamiento se espera aumentar a 100 autos para después expandirse a otras ciudades del país.



"Lo que hace este modelo es que las personas ocupen de manera más concientizada el vehículo por menores periodos de tiempo. Los estudios demuestran que las personas que forman parte del carsharing, aproximadamente un 30% decide renunciar al automóvil propio o decide vender el segundo automóvil", explica Francisco Loehnert, CEO de la compañía.

Para inscribirse es necesario ingresar a la página web, llenar un formulario y descargar la aplicación móvil. Como requisitos, es necesario contar con licencia de conducir al día y emitida con al menos dos años de antigüedad, buenos antecedentes comerciales y una tarjeta de crédito activa.

Las modalidad inicial para utilizar los autos es de modo "Round Trip", como lo es hoy en día 88% de los vehículos de car sharing en el mundo, en la que los autos se toman y se devuelven en el mismo punto de estacionamiento. Sin embargo, el equipo AWTO planea implementar la funcionalidad "Station Based", en la que los vehículos se pueden devolver en cualquier otro punto de la red, en los próximos meses, para lo cual se necesita una mayor red de estacionamientos y vehículos.

Existen dos planes de membresía mensual: Awto Ocasional, que tiene un valor de \$3.490 y Awto Frecuente, cuyo costo es de \$5.490, los cuales se descontarán a través de la tarjeta de crédito, al igual que el uso de cada viaje. El costo del arriendo del auto varía entre \$120 y \$150 por minuto manejado y de \$60 por minuto estacionado (bajo la reserva del usuario), según el tipo de membresía que se contrate; valores que incluyen bencina, estacionamiento, Tag, seguros, entre otros.

La reserva del auto se puede realizar vía teléfono, utilizando la aplicación o en www.awto.cl. El usuario recibirá un mail con la confirmación de la reserva, para luego de geo localizar el que esté más cerca del usuario, lo pueda utilizar por el tiempo que estime conveniente. Para desbloquear el vehículo, el usuario tan sólo tendrá que apretar el botón "abrir puertas" una vez que se encuentre frente al vehículo, desde la aplicación móvil.





A partir del caso anterior, conteste las siguientes preguntas:

Conteste las siguientes preguntas con contenidos vistos en clases. En caso contrario no recibirá puntaje.

1. Pensando en un Business Case: ¿Cuáles son los riesgos de un proyecto como AWTO?
2. ¿Debe o no debe ser privada la información de los suscriptores? Justifique.
3. Si Ud. fuera consultor contratado por AWTO para implementar un ERP, ¿cuáles serían los módulos que le recomendaría implementar?, ¿en qué orden? Adicionalmente, defina a lo menos 3 riesgos que enfrentaría el proyecto y sus respectivas mitigaciones.
4. Cada vez que un nuevo cliente de AWTO se inscribe el captador lo ingresa a una hoja de cálculos (Excel), y luego, al finalizar el día, todos los captadores envían un mail al área comercial para que esta cree una planilla de calculo final que consolida los datos entregados de cada captador. ¿Existe una mejor forma de manejar estos datos? ¿Qué le recomendaría y por qué?
5. El CEO de AWTO se reunió contigo, alumno destacado de la UAI, para preguntarte si crear o no una Gerencia de BI. Explícale qué problemas trae no tener un área de BI y que beneficios trae tener un área de BI.
6. Explique al menos tres Sistemas de Información que usa Dafiti, qué área lo usa, cómo lo usa, y cual es el valor que entrega a esa área.

1.- Un proyecto como AWTO al poseer S.I. corre riesgos como: falla en el software de la App (usuario), falla en el mantenimiento (por parte de AWTO), que se filtre información privada que se maneje y que se caiga el sistema de la empresa.





2- Por supuesto, la información obtenida de los suscriptores debe ser privada, a menos que el cliente autorice que sus datos puedan ser usados con algún otro propósito pero para ello se debe firmar un acuerdo de confidencialidad entre las partes. De otro modo cualquier otro uso o filtración de datos privados de los suscriptores puede ser penado por la ley y caer en demandas para la empresa.

3- Como consultor le recomendaría implementar un ERP para los módulos de: ~~Arrendados~~ ~~Arrendados~~

- Administración de datos, respecto a los suscriptores. Para lograr hacer un uso eficiente de dichos datos.
- Respecto a la flota de Auto con la que cuenta Auto, con todos los aspectos de cada uno (Km, consumo, seguros, usuarios, etc..)

Entre los riesgos que enfrenta el proyecto y sus mitigaciones

- | Riesgo | Mitigación |
|------------------------------|--|
| 1) Duplicación de datos | 1- ERP, evitaría este problema |
| 2) Bajas ventas (Arrendados) | 1- CRM, para conocer de mejor forma al cliente |
| 3) Filtración de datos | 1- S.I. que permita cifrar datos confidenciales. |





4. Sí, Existen mejores formas de administrar los Datos. Para ello las medianas y grandes empresas usan algún tipo de ERP (SAP, Oracle, etc.) que les permiten hacer un uso más eficiente de los datos y ~~administrarlos~~ administrarlos de mejor manera. ~~Datos duplicados~~ ~~Datos duplicados~~

En este caso recomendaría un ERP para medianas empresas, de esa manera se evitarían riesgos que está corriendo la empresa como: Datos Duplicados, inconsistencia de datos y largos periodos de trabajo para un proceso que se puede automatizar con un Sistema de información.

5- Dentro de los problemas que trae no tener B.I:

- Malas estimaciones de futuros usuarios.
- Desconocimiento de lo que está ocurriendo en el mercado
- Malas decisiones respecto al futuro de la empresa.





Entre los beneficios que trae tener un B.I.

- Se posee respaldo de las decisiones
- Se pueden hacer estimaciones sólidas y fundamentadas.
- Se tiene conocimiento del mercado

Por lo tanto es bueno tener experiencia, pero es mejor tener B.I. para la toma de decisiones. ~~Se recomienda tener un área de B.I., aún más para la industria que se está hablando.~~

6.-

1) Tablau: ~~corresponde~~ Lo utiliza en el área de post-venta y lo utiliza para llevar un respaldo de ~~todos~~ todos los aspectos respecto a la venta. Además le entrega un gran valor a esa área dado su capacidad y eficacia para el manejo de datos.

2) WMS: Lo utiliza en el área de proveer ~~datos~~ y lleva un registro acabado de toda la logística que fue necesaria para la compra del mismo.

3) OMS: Se utiliza en el área de ventas, ~~pero~~ pero integra todas las áreas de ~~ventas~~ ~~ventas~~ permitiendo tener una de las mejores logísticas de Chile.



C.8 Rúbrica pruebas 1 sistemas de información

P1	Respuesta incorrecta o no responde 0puntos	Hace una descripción sin riesgos claros 2puntos	Detecta riesgos sin mayores detalles 4puntos	Detecta riesgos y explica a la perfección cada uno 6puntos		
P2	Respuesta incorrecta o no responde 0puntos	La información puede ser privada o no. Si ocurre uno de los dos siguientes casos: - Indica vagamente qué información, pero no justifica nada - justifica vagamente que debe ser privada o no privada, pero sin dejar claramente sobre qué información lo está realizando 2puntos	La información puede ser privada o no. Si ocurre uno de los dos siguientes casos: - Indica claramente (muy bien descrito y bien contextualizada) qué información es la que se considerará privada o no privada, pero no justifica nada - justifica claramente que debe ser privada o no privada y los argumentos son coherentes al problema, pero sin dejar claramente sobre qué información en concreto está realizando su justificación 4puntos	La información puede ser o no privada, debe entregar la justificación de la privacidad o la no privacidad indicando al menos uno de estos dos casos: - PRIVADA: cuál información que se considera privada y por qué se considera privada (qué vulnera algo en el suscriptor del servicio); - NO PRIVADA: cuál es la información sobre la cual realizará la justificación y por qué no se considera privada 6puntos		
P3	Respuesta incorrecta o no responde 0puntos	Responde módulos del ERP que no tienen que ver con caso y no explica con una cierta lógica la relación estos módulos con los datos del caso. Nombra riesgos, pero no los explica, o los explica sin relación al caso 2puntos	Nombra 1 Inventarios, 2 Ventas, 3 E Business, 4 Contabilidad, 5 CRM, 6 BI, pero no los explica en relación al caso. o Nombra Riesgos, Resistencia de Personas, Procesos mal implementados, grandes costos de implementación, organización no preparada para el cambio, mala definición de requerimientos, pero no los explica en relación al caso. O los nombra y explica pero no da Mitigaciones 4puntos	Nombra 1 Inventarios, 2 Ventas, 3 E Business, 4 Contabilidad, 5 CRM, 6 BI, los explica en relación al caso. Nombra Riesgos, Resistencia de Personas, Procesos mal implementados, grandes costos de implementación, organización no preparada para el cambio, mala definición de requerimientos, los explica en relación al caso, y explica posibles Mitigaciones para los 3 riesgos pedidos 6puntos		
P4	Respuesta incorrecta o no responde 0puntos	Explica una de las cinco etapas: que mediante un SI se puede crear un ciclo de datos efectivo, por el cual se ingresan datos (fuentes) a la base de datos única (data warehouse) y luego los analistas crean sus bases de datos intermedias (datamarts) para luego hacerles consultas o minería (queries y mining) para finalmente darle soluciones a otras áreas 1.2puntos	Explica dos de las cinco etapas: que mediante un SI se puede crear un ciclo de datos efectivo, por el cual se ingresan datos (fuentes) a la base de datos única (data warehouse) y luego los analistas crean sus bases de datos intermedias (datamarts) para luego hacerles consultas o minería (queries y mining) para finalmente darle soluciones a otras áreas 2.4puntos	Explica tres de las cinco etapas: que mediante un SI se puede crear un ciclo de datos efectivo, por el cual se ingresan datos (fuentes) a la base de datos única (data warehouse) y luego los analistas crean sus bases de datos intermedias (datamarts) para luego hacerles consultas o minería (queries y mining) para finalmente darle soluciones a otras áreas 3.6puntos	Explica cuatro de las cinco etapas: que mediante un SI se puede crear un ciclo de datos efectivo, por el cual se ingresan datos (fuentes) a la base de datos única (data warehouse) y luego los analistas crean sus bases de datos intermedias (datamarts) para luego hacerles consultas o minería (queries y mining) para finalmente darle soluciones a otras áreas 4.8puntos	Explica que mediante un SI se puede crear un ciclo de datos efectivo, por el cual se ingresan datos (fuentes) a la base de datos única (data warehouse) y luego los analistas crean sus bases de datos intermedias (datamarts) para luego hacerles consultas o minería (queries y mining) para finalmente darle soluciones a otras áreas 6puntos
P5	Respuesta incorrecta o no responde 0puntos	Explica solo los problemas de no tener una área de BI 2puntos	Explica solo los beneficios 4puntos	Explica que al no tener un área de BI muchas decisiones las toman "expertos" que deben estar disponibles (contratados) y las decisiones deben tomarse por experiencia y de forma subjetiva. Los beneficios son, la mejora o agilidad organizacional, empoderar a las personas en la compañía, mejores decisiones, y lo anterior hace que los clientes estén más contentos. 6puntos		
P6	Respuesta incorrecta o no responde 0puntos	Describe algo general pero sin abordar sistemas específicos 2puntos	Entrega toda la información de 2 sistemas o menos 4puntos	Entrega toda la información de forma correcta de tres sistemas 6puntos		

C.9 Pruebas 2 sistemas de información



PRUEBA 2

NOMBRE

RUT: 17586810

CURSO: SISTEMAS DE INFORMACIÓN SEC.4 STGO FIC STGO 2017/1

PÁGINA: 1 DE 5



Sistemas de Información

Prueba 2

Feltrinelli Group es una polifacética compañía italiana que abarca un abanico de actividades comerciales muy diversas. Fundada en Italia originalmente como una editorial en 1954, cuenta con aproximadamente 2000 empleados y ostenta ingresos anuales superiores a 350 millones de euros. Sus actividades principales se centran en la gestión de la cadena de librerías más grande de Italia (alrededor de 120 tiendas) y de su página web, que presta servicios a una comunidad de más de dos millones de fieles clientes italianos y extranjeros. El grupo también cuenta con una editorial de fama mundial, una cadena de restaurantes, un canal de televisión y una agencia inmobiliaria.

La gestión de los datos generados por un grupo de empresas tan complejo siempre es un desafío, en el mejor de los casos. Sin embargo, en 2014, se hizo evidente que los sistemas de información de gestión existentes debían renovarse con urgencia. En la vertiginosa actualidad, el acceso a datos de buena calidad en tiempo real es un factor fundamental para el éxito continuo de las empresas, pero esto es algo que Feltrinelli tenía dificultad para lograr. Raffaele Gamberini, CIO de Feltrinelli, explica lo siguiente:

desafío en
acceso a
datos de
buena calidad

"En algunas áreas todavía usábamos sistemas que habían sido implementados antes del cambio de milenio, y estaba claro que el uso continuado de estos sistemas estaba afectando a la eficiencia de nuestros negocios. Necesitábamos una solución que nos permitiera rediseñar completamente nuestra forma de trabajar".

problema

Empresa: Feltrinelli Group

Industria: Servicios/Minorista

Desafíos

- Mejorar, simplificar y optimizar procesos comerciales
- Modernizar sistemas informáticos
- Preparar a su gente para el nuevo desafío
- Gestionar como un Holding (Grupo) la empresa

Beneficios Buscados:

- Tranquilidad (confiabilidad en los resultados) → B.I
- Plataforma flexible que se desarrollará según las necesidades cambiantes → SCM
- Rápido retorno de la inversión
- Mejor costo de transacción
- Integración de áreas y empresas del grupo → ERP
- Relaciones de Largo plazo con Clientes y proveedores
- Foco en sus clientes → CRM
- Robusto proceso de planificación
- Lealtad de sus clientes





PRUEBA 2

NOMBRE

RUT: 17586810

CURSO: SISTEMAS DE INFORMACIÓN SEC.4 STGO FIC STGO 2017/1

Contiene preguntas con contenidos vistos en clases. En caso contrario recibirá puntaje. Justifique todas sus respuestas.



✓ Pregunta 1: (1 punto)

Feltrinelli Group (FG) tiene un Departamento de Operaciones (DO), que dentro de sus trabajos está monitorear constantemente el Stock de libros de todas las editoriales. Hoy tienen 10 analistas enfocados solo en esta tarea y todos deben conversar con distintos mayoristas que venden los libros de las editoriales. ¿Existe algún SI que podría ayudar a FG en esta tarea? Justifique su respuesta indicando costos y beneficios.

✓ Pregunta 2: (1 punto)

Feltrinelli Group (FG) tiene agentes de ventas que visitan colegios para hacer ventas masivas. Estos agentes son vendedores especializados que llevan años trabajando en la industria, pero los sueldos son tan bajos que la rotación de estos entre otras librerías de Italia es alta. El gran problema que tiene FG es que cada vez que renuncia un agente se lleva la información de todo el trabajo de venta realizado, ya que lo tienen todo en agendas y cuadernos. ¿Qué SI recomendaría a FG para resolver este problema? Justifique su respuesta indicando costos y beneficios.

✓ Pregunta 3: (1 punto)

- A. (0,5 puntos) Haga un diagnóstico de la situación de la empresa, defina porque no puede seguir operando con sus sistemas actuales y proponga una solución para un rápido retorno de la inversión y generar lealtad de sus clientes. Justifique todos sus supuestos.
- B. (0,5 puntos) Dado el diagnóstico realizado, proponga una solución de SI que permita al grupo Feltrinelli, entregar a sus empleados información de beneficios, trámites y procesos de personal. Además, proponga una forma de Ecommerce que permita a los empleados comprar productos de la compañía en forma fácil. Justifique todos sus supuestos.

Pregunta 4: (1 punto)

Si se implementa un CRM para el Grupo Feltrinelli, ¿con cuáles sistemas actuales del Grupo Feltrinelli debiera integrarse? ¿Por qué?

Pregunta 5: (1 punto)

¿Cómo sería una solución de Analítica Predictiva (Business Analytics) para el Grupo Feltrinelli? Debe responder indicando las 5 preguntas de negocio que la solución indicada por usted puede responder para el Grupo Feltrinelli.

Pregunta 6: (1 punto)

De Acuerdo a la Charla, ¿cuáles son los pro y los contra de un ERP 100% en la nube versus un ERP que opera en un Data Center Tradicional?. Además, explique cuáles son las responsabilidades éticas de la empresa Defontana con respecto a la confidencialidad de la información de sus clientes.





1) Problema

↓
• Gestión de inventario

— Si existe S.I que puedan a mejorar el proceso de Gestión de inventario como lo son los ~~SCM~~ SCM (System content Management), este S.I permitiría a los ~~usuarios~~ usuarios ir modificando la información respecto las ventas que ~~se~~ se van ~~efectuando~~ efectuando, de esa forma los analistas solo debiesen ir chequeando la información actualizada en la plataforma de la compañía.

Entre los costos que implicaría encontramos: costos de ~~capacitación~~ capacitación para empleados, licencias, mantención del software.
y entre los beneficios: disminución en tiempos para tener la información (ventas), optimación de procesos y disminución de HH.

2) Los S.I que se recomienda son dos, y Primero un sistema CRM para que se tenga estandarizado la estrategia con la que se capturan clientes y en Segundo lugar un ERP para Almacenar toda información en una "hubo" de la compañía, de esta manera se evitaría que los vendedores se fueran con la información, puesto que esta estaría respaldada.

Entre los

costos

- Licencia para la ERP
- Membresía de ERP
- Capacitación a los vendedores

Beneficios

- seguridad/respaldo de los datos
- Misma política de captación de clientes ~~por vendedores~~





3) ^{3.5} Las alta rotación de vendedores, quienes se llevan la información ~~con ellos~~ de venta con ellos esta perjudicando fuertemente a la empresa, ya que ~~esta~~ dicha información pertenece a la empresa. Por ello es necesario implementar un sistema de ERP en esta situación para lograr respaldar la información respecto a las ventas. Por otro lado el trabajo tedioso de recopilar la información por parte de los analistas con respecto a los mayorista, se ~~podría~~ podría mejorar drásticamente con un sistema de SCM que permita ir registrando la información de una manera más eficiente y limpia. Estas medidas permitirían modernizar los S.I. además de solucionar 2 Áreas significativas de la empresa, mejorando la lealtad de los clientes, puesto que su información respecto a la compra anterior no se perdería, ya que estaría respaldada y cualquier trabajador podría solucionar sus problemas.

3.6

Una solución S.I que permita al grupo Feltrinelli entregar información a sus empleados como (beneficios, trámites y procesos personales) sería un SCM, como un estilo de intranet para empleado, donde ellos pudiesen ingresar con su rut y les permita agregar/modificar información como la anteriormente señalada.

Una forma de E-commerce, sería E2C (Employee - company) como lo es la intranet de la universidad (Uki) donde esta la opción de pagar distintos tipos de certificados/documentos como ~~Nota~~: Rivalidación de pase escolar.





4) En primer lugar debiese integrarse con el sistema de ventas que tiene actualmente la compañía, aunque se recomienda que dicho sistema también debiese modernizarse (Preg 2). Además debiese integrarse con el departamento de operaciones con el propósito de estar alineado con el stock que posee la compañía. Esto porque cualquier mejora o modernización de proceso debe considerar poder integrarse de la mejor manera con las distintas áreas de la compañía para permanecer alineada con esta.

- 5)
- 1) ¿Qué datos tengo?
 - 2) ¿Qué señalan estos?
 - 3) ¿Qué decidido?
 - 4) ¿proyecciones?
 - 5) ¿cómo mejoró?

6) Los pro de tener el ERP 100% en la nube, son ~~su~~ su capacidad de respuesta ante incidentes naturales, como el terremoto del 2010 en donde las empresas que tenían su información en la nube no tuvieron que preocuparse por pérdidas de datos por daños físicos a la computadora. Por otro lado se corre el riesgo ^(contra) a ataques cibernéticos por terroristas, poniendo en riesgo a la compañía, además de posibles errores humanos en el manejo de información (filtración de datos).

Doton lana posee un deber ético por velar del bienestar y ~~estabilidad~~ estabilidad de la información de la empresa, cualquier mal uso de dicha información la empresa puede caer en actos legales contra Doton lana.



C.10 Rúbrica pruebas 2 sistemas de información

P1	Respuesta incorrecta o no responde 0puntos	Menciona dos áreas y explica bien, o al menos tres de forma incompleta 2puntos	Menciona tres áreas y explica bien, o al menos cuatro de forma incompleta 4puntos	Explica cinco áreas y explica de forma completa 6puntos	
P2	Respuesta incorrecta o no responde 0puntos	Menciona una desventaja y una ventaja de forma incompleta 2puntos	Menciona una desventaja y una ventaja de forma completa, o bien dos ventajas y dos desventajas de forma incompleta 4puntos	Menciona dos desventajas y dos ventajas de forma completa 6puntos	
P3	Respuesta incorrecta o no responde 0puntos	responde solo 3 requerimientos los que pueden ser RF o RNF o RD (se consideran máximo 3 de cada uno de los tipos de requerimientos) 2puntos	responde solo 6 requerimientos los que pueden ser RF o RNF o RD (se consideran máximo 3 de cada uno de los tipos de requerimientos) 4puntos	responde los 3 RF, 3 RNF y los 3 RD (se consideran máximo 3 de cada uno de los tipos de requerimientos) 6puntos	
P4	Respuesta incorrecta o no responde 0puntos	da un argumento correcto (no importa si es a favor o en contra debe ser coherente con la postura) 2puntos	da dos argumentos correctos (no importa si es a favor o en contra debe ser coherente con la postura) 4puntos	da tres argumentos correctos (no importa si es a favor o en contra debe ser coherente con la postura) 6puntos	
P5	Respuesta incorrecta o no responde 0puntos	Responde 1 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 1.5puntos	Responde 2 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 3puntos	Responde 3 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 4.5puntos	Responde 4 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 6puntos
P6	Respuesta incorrecta o no responde 0puntos	Responde en forma parcial las 3 preguntas de como TI se debe administrar en la Organización o responde en forma parcial el modelo de gobierno de Monarquía TI para soporte básico de TI en la empresa 2puntos	Responde en forma completa las 3 preguntas de como TI se debe administrar en la Organización O responde en forma Completa el modelo de gobierno de Monarquía TI para soporte básico de TI en la empresa (explica a lo menos una de ellas) 4puntos	Responde en forma completa las 3 preguntas de como se administra TI en la Organización (como se organiza TI para dar soporte local y global, que habilidades se mantienen y cuales se deben "Outsourcear" y como se desarrollan los proyectos y con que métricas) Y responde en forma completa el modelo de gobierno de Monarquía TI para soporte básico u otro modelo compatible (explica ambas) 6puntos	
P7	Respuesta incorrecta o no responde 0puntos	comenta que las TI apoyan al proceso productivo o a las operaciones mineras, pero no hace referencia a las etapas y no comenta de ningún sistema 2puntos	comenta que las TI apoyan al proceso productivo o a las operaciones mineras, hace referencia al menos a 2 etapas del proceso productivo o comenta de sistemas que calzan en al menos dos de las etapas del proceso productivo 4puntos	las TI en la industria minera están apoyando no sólo a la parte administrativa y de oficina, sino que también al mundo productivo, es decir directamente a las operaciones mineras, en sus distintas etapas del proceso productivo: exploración, planificación, extracción, procesamiento y comercialización. Slide 18 de la presentación del invitado a la charla 3. Puede también mencionar sistemas de las etapas anteriormente indicadas y es equivalente. 6puntos	

C.11 Pruebas 3 sistemas de información

Se utilizó Emarking solo para hojas de respuestas, no para generar enunciado. Prueba con 6 preguntas y 4 hojas para respuesta.

C.12 Rúbrica pruebas 3 sistemas de información

P1	Respuesta incorrecta o no responde 0puntos	Menciona dos áreas y explica bien, o al menos tres de forma incompleta 2puntos	Menciona tres áreas y explica bien, o al menos cuatro de forma incompleta 4puntos	Explica cinco áreas y explica de forma completa 6puntos	
P2	Respuesta incorrecta o no responde 0puntos	Menciona una desventaja y una ventaja de forma incompleta 2puntos	Menciona una desventaja y una ventaja de forma completa, o bien dos desventajas de forma incompleta 4puntos	Menciona dos desventajas y dos ventajas de forma completa 6puntos	
P3	Respuesta incorrecta o no responde 0puntos	responde solo 3 requerimientos los que pueden ser RF o RNF o RD (se consideran máximo 3 de cada uno de los tipos de requerimientos) 2puntos	responde solo 6 requerimientos los que pueden ser RF o RNF o RD (se consideran máximo 3 de cada uno de los tipos de requerimientos) 4puntos	responde los 3 RF, 3 RNF y los 3 RD (se consideran máximo 3 de cada uno de los tipos de requerimientos) 6puntos	
P4	Respuesta incorrecta o no responde 0puntos	da un argumento correcto (no importa si es a favor o en contra debe ser coherente con la postura) 2puntos	da dos argumentos correctos (no importa si es a favor o en contra debe ser coherente con la postura) 4puntos	da tres argumentos correctos (no importa si es a favor o en contra debe ser coherente con la postura) 6puntos	
P5	Respuesta incorrecta o no responde 0puntos	Responde 1 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 1.5puntos	Responde 2 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 3puntos	Responde 3 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 4.5puntos	Responde 4 de los siguientes 5 puntos: Privacy and Security Challenges Governance, SLAs and QoS Reliability and Credibility Cloud Integration and Interoperability Cross-Cloud Composition, Collaboration and Coordination Applications 6puntos
P6	Respuesta incorrecta o no responde 0puntos	Responde en forma parcial las 3 preguntas de como TI se debe administrar en la Organización o responde en forma parcial el modelo de gobierno de Monarquía TI para soporte básico de TI en la empresa 2puntos	Responde en forma completa las 3 preguntas de como TI se debe administrar en la Organización O responde en forma Completa el modelo de gobierno de Monarquía TI para soporte básico de TI en la empresa (explica a lo menos una de ellas) 4puntos	Responde en forma completa las 3 preguntas de como se administra TI en la Organización (como se organiza TI para dar soporte local y global, que habilidades se mantienen y cuales se deben "Outsourcear" y como se desarrollan los proyectos y con que metricas) Y responde en forma completa el modelo de gobierno de Monarquía TI para soporte básico u otro modelo compatible (explica ambas) 6puntos	
P7	Respuesta incorrecta o no responde 0puntos	comenta que las TI apoyan al proceso productivo o a las operaciones mineras, pero no hace referencia a las etapas y no comenta de ningún sistema 2puntos	comenta que las TI apoyan al proceso productivo o a las operaciones mineras, hace referencia al menos a 2 etapas del proceso productivo o comenta de sistemas que calzan en al menos dos de las etapas del proceso productivo 4puntos	las TI en la industria minera están apoyando no sólo a la parte administrativa y de oficina, sino que también al mundo productivo, es decir directamente a las operaciones mineras, en sus distintas etapas del proceso productivo: exploración, planificación, extracción, procesamiento y comercialización. Slide 18 de la presentación del invitado a la charla 3. Puede también mencionar sistemas de las etapas anteriormente indicadas y es equivalente. 6puntos	

D Script extracción de datos desde base de datos

```
1. <?php
2. require_once(dirname(dirname(__FILE__))."/config.php");
3. require_once($CFG->dirroot."/lib/csvlib.class.php");
4. require_once ($CFG->dirroot . "/mod/emarking/locallib.php");
5. global $DB, $CFG;
6.
7. $course = required_param("course", PARAM_TEXT);
8. $num = required_param("num", PARAM_INT);
9. $puntajemaximo = optional_param("max", 0, PARAM_INT);
10. // programacion prueba 0
11. $progra0 = array(
12.     825229,
13.     825230,
14.     825233,
15.     825235
16. );
17. // programacion prueba 1
18. $progra1 = array(
19.     837371,
20.     837372,
21.     837375,
22.     837377
23. );
24. // programacion examen
25. $prograexamen = array(
26.     846505,
27.     846504,
28.     846508,
29.     846510
30. );
31. // sistemas prueba 1
32. $sist1 = array(
33.     825844,
34.     825847,
35.     825854,
36.     825855
37. );
38. // sistemas prueba 2
39. $sist2 = array(
40.     833504,
41.     833503,
42.     833501,
43.     833500
44. );
45. // sistemas prueba 3
46. $sist3 = array(
47.     845129,
48.     845132,
49.     845134,
50.     845133
51. );
52. $csv = new csv_export_writer("semicolon");
53. $head = array(
```

```

54.         'commentid',
55.         'courseid',
56.         'emarkingid',
57.         'emarkingname',
58.         'markerid',
59.         'markername',
60.         'studentid',
61.         'submissionid',
62.         'draftid',
63.         'score',
64.         'bonus',
65.         'length',
66.         'text',
67.         'links'
68.     );
69.     $csv->add_data($head);
70.     switch ($course) {
71.         case 'progra':
72.             switch ($num) {
73.                 case 0:
74.                     $option = $progra0;
75.                     break;
76.                 case 1:
77.                     $option = $progra1;
78.                     break;
79.                 case 2:
80.                     $option = $prograexamen;
81.                     break;
82.             }
83.             break;
84.         case 'sist':
85.             switch ($num) {
86.                 case 0:
87.                     $option = $sist1;
88.                     break;
89.                 case 1:
90.                     $option = $sist2;
91.                     break;
92.                 case 2:
93.                     $option = $sist3;
94.                     break;
95.             }
96.             break;
97.     }
98.     foreach ($option as $prueba){
99.         if (!$cm = get_coursemodule_from_id('emarking', $pr
ueba)) {
100.             print_error(get_string('invalidcoursemodule
', 'mod_emarking') . " id: $cmid");
101.         }
102.         // Validate eMarking activity //TODO: validar draft
si está selccionado.
103.         if (!$emarking = $DB->get_record('emarking', array(
104.             'id' => $cm->instance
105.         ))) {
106.             print_error(get_string('invalidid', 'mod_em
arking') . " id: $cm->id");

```

```

107.         }
108.
109.         $allstudents = emarking_get_students_for_printing($
cm->course);
110.         $list = '(';
111.         foreach ($allstudents as $student){
112.             $list .= $student->id.',';
113.         }
114.         $list = rtrim($list, ',');
115.         $list .= ')';
116.         // Sin enhanced feedback y sin puntajes maximos
117.         $moreoptions1 = "";
118.         $moreoptions2 = "";
119.         if($puntajemaximo){
120.             $moreoptions1 = "inner join (select
121.                             r.id,
r.criterionid,
122.                             rr.maxscore
123.                             from
mdl_gradingform_rubric_levels as r
124.                             inner join(select
125.
id,
126.
criterionid,
127.
max(score) as
128.
maxscore
129.
from mdl_gradingform_rubric_levels
130.
group by criterionid) as rr on (r.criterionid =
rr.criterionid) group by id)
131.                             as sc on (sc.id =
c.levelid)";
132.             $moreoptions2 = "and (r.score+c.bonus) <
sc.maxscore";
133.         }
134.         // sin links oer y por defecto con puntajes maximos
135.         $sql = "select
136.                 c.id as commentid,
137.                 e.course,
138.                 e.id as emarkingid,
139.                 e.name as emarkingname,
140.                 u.id as markerid,
141.                 concat(u.firstname, ' ', u.lastname) as
alumno,
142.                 s.student as studentid,
143.                 s.id as submissionid,
144.                 d.id as draftid,
145.                 r.score,
146.                 c.bonus,
147.                 length(c.rawtext),
148.                 c.rawtext,
149.                 '0'
150.                 from mdl_emarking as e

```

```

151.         inner join mdl_emarking_submission as s on
           (e.id = s.emarking)
152.         inner join mdl_emarking_draft as d on (s.id
           = d.submissionid)
153.         inner join mdl_emarking_comment as c on
           (c.draft = d.id)
154.         inner join mdl_user as u on (s.student =
           u.id)
155.         inner join mdl_gradingform_rubric_levels as
           r on (r.id = c.levelid)
156.         $moreoptions1
157.         where e.id = $emarking->id
158.         and s.student in $list
159.         AND c.id not in (select
160.                             c.i
161.                             d
162.                             from
163.                             m mdl_emarking as e
164.                             er join mdl_emarking_submission as s on (e.id = s.emarking)
165.                             er join mdl_emarking_draft as d on (s.id = d.submissionid)
166.                             er join mdl_emarking_comment as c on (c.draft = d.id)
167.                             er join mdl_emarking_feedback as f on (c.id = f.commentid))
168.         $moreoptions2
169.         group by c.id
170.         order by marker";
171.         if($data = $DB->get_records_sql($sql)){
172.             foreach ($data as $row) {
173.                 $newdata = array();
174.                 foreach ($row as $field){
175.                     $newdata[] = $field
176.                 }
177.                 $csv->add_data($newdata);
178.             }
179.             $sql = "select
180.                     c.id as commentid,
181.                     e.course,
182.                     e.id as emarkingid,
183.                     e.name as emarkingname,
184.                     u.id as markerid,
185.                     concat(u.firstname, ' ',
186.                             u.lastname) as alumno,
187.                     s.student as studentid,
188.                     s.id as submissionid,
189.                     d.id as draftid,
190.                     r.score,
191.                     c.bonus,
192.                     length(c.rawtext),
193.                     c.rawtext,
194.                     count(f.id)
195.                     from mdl_emarking as e

```

```

194.             inner join
195.             mdl_emarking_submission as s on (e.id = s.emarking)
196.             inner join
197.             mdl_emarking_draft as d on (s.id = d.submissionid)
198.             inner join
199.             mdl_emarking_comment as c on (c.draft = d.id)
200.             inner join
201.             mdl_emarking_feedback as f on (c.id = f.commentid)
202.             inner join mdl_user as u on
203.             (s.student = u.id)
204.             inner join
205.             mdl_gradingform_rubric_levels as r on (r.id = c.levelid)
206.             where e.id = $emarking->id
207.             group by c.id order
208.             by username";
209.             if($data = $DB->get_records_sql($sql)){
210.                 foreach ($data as $row) {
211.                     $newdata = array();
212.                     foreach ($row as $field){
213.                         $newdata[] = $field
214.                     }
215.                 }
216.             }
217.             $csv->add_data($newdata);
218.         }
219.     }
220.     $csv->set_filename('Data_'.$course.'_'. $num.'_'.time());
221.     $csv->download_file();

```

```

1. <?php
2. require_once(dirname(dirname(__FILE__))."/config.php");
3. require_once($CFG->dirroot."/lib/csvlib.class.php");
4. require_once ($CFG->dirroot . "/mod/emarking/locallib.php");
5. global $DB, $CFG;
6.
7. $course = required_param("course", PARAM_TEXT);
8. $num = required_param("num", PARAM_INT);
9. $sess = optional_param("sess", 1, PARAM_INT);
10. // programacion prueba 0
11. $progra0 = array(
12.     825229,
13.     825230,
14.     825233,
15.     825235
16. );
17. // programacion prueba 1
18. $progra1 = array(
19.     837371,
20.     837372,
21.     837375,
22.     837377
23. );
24. // programacion examen
25. $prograexamen = array(

```

```

26.             846505,
27.             846504,
28.             846508,
29.             846510
30.     );
31.     // sistemas prueba 1
32.     $sist1 = array(
33.         825844,
34.         825847,
35.         825854,
36.         825855
37.     );
38.     // sistemas prueba 2
39.     $sist2 = array(
40.         833504,
41.         833503,
42.         833501,
43.         833500
44.     );
45.     // sistemas prueba 3
46.     $sist3 = array(
47.         845129,
48.         845132,
49.         845134,
50.         845133
51.     );
52.     // CSV sesiones
53.     $csvsessions = new csv_export_writer("semicolon");
54.     $head = array(
55.         'sessid',
56.         'firstname',
57.         'lastname',
58.         'idnumber',
59.         'userid',
60.         'draftid',
61.         'starttime',
62.         'endtime'
63.     );
64.     $csvsessions->add_data($head);
65.     switch ($course) {
66.         case 'progra':
67.             switch ($num) {
68.                 case 0:
69.                     $option = $progra0;
70.                     break;
71.                 case 1:
72.                     $option = $progra1;
73.                     break;
74.                 case 2:
75.                     $option = $prograexamen;
76.                     break;
77.             }
78.             break;
79.         case 'sist':
80.             switch ($num) {
81.                 case 0:
82.                     $option = $sist1;

```

```

83.                                     break;
84.                                     case 1:
85.                                         $option = $sist2;
86.                                         break;
87.                                     case 2:
88.                                         $option = $sist3;
89.                                         break;
90.                                     }
91.                                     break;
92.     }
93.
94.     foreach ($option as $prueba){
95.         if (!$cm = get_coursemodule_from_id('emarking', $pr
ueba)) {
96.             print_error(get_string('invalidcoursemodule
', 'mod_emarking') . " id: $cmid");
97.         }
98.         // Validate eMarking activity //TODO: validar draft
si está selccionado.
99.         if (!$emarking = $DB->get_record('emarking', array(
100.             'id' => $cm->instance
101.         ))) {
102.             print_error(get_string('invalidid', 'mod_em
arking') . " id: $cm->id");
103.         }
104.
105.         $allstudents = emarking_get_students_for_printing($
cm->course);
106.         $list = '(';
107.         foreach ($allstudents as $student){
108.             $list .= $student->id.',';
109.         }
110.         $list = rtrim($list, ',');
111.         $list .= ')';
112.         // sessions
113.         $sqlsessions = "SELECT
114.                         sess.id,
115.                         u.firstname,
116.                         u.lastname,
117.                         u.idnumber,
118.                         userid,
119.                         draftid,
120.                         sess.starttime,
121.                         sess.endtime
122.                         FROM {emarking} AS e INNER JOIN
{emarking_draft} AS draft ON (e.id = ? AND e.id =
draft.emarkingid)
123.                         INNER JOIN {emarking_session} AS
sess ON (sess.draftid = draft.id AND sess.userid IN $list)
124.                         INNER JOIN {user} AS u ON (u.id =
sess.userid)";
125.         if($data = $DB-
>get_records_sql($sqlsessions, array($emarking->id)){
126.             foreach ($data as $row) {
127.                 $newdata = array();
128.                 foreach ($row as $field){
129.                     $newdata[] = $field;

```



```

130.     }
131.     $csvsessions->add_data($newdata);
132.     }
133. }
134. // eval feedback
135. $head = array(
136.     'evalid',
137.     'firstname',
138.     'lastname',
139.     'idnumber',
140.     'userid',
141.     'draftid',
142.     'complexity',
143.     'relevant',
144.     'personalization',
145.     'timecreated',
146.     'lastmodified',
147.     'optionalcomment'
148. );
149. $csvsessions->add_data($head);
150. $sqleval = "SELECT
151.     eval.id,
152.     u.firstname,
153.     u.lastname,
154.     u.idnumber,
155.     eval.userid,
156.     eval.submissionid,
157.     eval.complexity,
158.     eval.relevant,
159.     eval.personalization,
160.     eval.timecreated,
161.     eval.lastmodified,
162.     eval.optionalcomment
163. FROM {emarking} AS e INNER JOIN
    {emarking_draft} AS draft ON (e.id = ? AND e.id =
    draft.emarkingid)
164.     INNER JOIN
    {emarking_evaluatefeedback} AS eval ON (eval.submissionid =
    draft.id AND eval.userid IN $list)
165.     INNER JOIN {user} AS u ON (u.id =
    eval.userid)";
166.     if($data = $DB-
    >get_records_sql($sqleval, array($emarking->id)){
167.         foreach ($data as $row) {
168.             $newdata = array();
169.             foreach ($row as $field){
170.                 $newdata[] = $field;
171.             }
172.             $csvsessions->add_data($newdata);
173.         }
174.     }
175. }
176.
177. $csvsessions-
    >set_filename('SESSIONS_'.$course.'_'. $num.'_'.time());
178. $csvsessions->download_file();

```

F Presentación de capacitación correctores

Feedback Formativo

Hans Jeria

Definición

El objetivo del feedback formativo es, sin importar si es entregado por un profesor o una computadora, en la sala de clases o en cualquier lugar, es **mejorar el aprendizaje y/o rendimiento**, generando la información adecuada, dirigida a conceptualizaciones y habilidades.

*Albertson, 1986; Azevedo & Bernard, 1995; Narciss & Hulth, 2004;
VanLehn, 1982.*

Funciones



Directiva

Le cuenta al estudiante que necesita arreglar o revisar. El feedback debe ser específico.

Facilitadora

Provee comentarios y sugerencias que ayudan a guiar al estudiante en su propia revisión y conceptualización.

Tipos de información



Verificación

Confirma si la pregunta es correcta o incorrecta, además, señala la escala de medición. Ejemplo la rúbrica.

Elaboración

Corresponde a entregar la respuesta correcta, elaborar sobre el tema central, discutir el error, proveer de ejemplos o una guía de resolución.

Framework

Framework se resume en

1. Cosas que hacer.
2. Cosas que evitar.
3. Qué hacer dependiendo del tiempo.
4. Qué hacer respecto a las características del alumno.

Para este experimento los puntos 3 y 4 están dados por el contexto, es decir el feedback no será inmediato eso significa que es una buena opción cuando se desea transferir conocimiento. Por otro lado, el punto cuatro apunta en particular a cada estudiante y su motivación por aprender en cada curso.

Cosas que hacer

- Atención en el feedback no en el estudiante.
- Proveer feedback elaborado para mejorar el aprendizaje del estudiante. El feedback debe describir que, como y/o porque algo está malo o incompleto.
- Se específico y claro con los mensajes del feedback. Puede frustrar a los estudiantes. En lo posible ser específico con el resultado esperado.*
- Mantén el feedback simple pero no tan simple. Generalmente provee suficiente información para ayudar al estudiante y no más. Un feedback complejo no genera un mayor aprendizaje.
- Reduce la incertidumbre entre rendimiento y metas.

Cosas que evitar

- No hacer comparaciones entre estudiantes, implícitas o explícitas.
- No realizar comentarios que desalienten al alumno o amenacen su autoestima. Usar el sentido común.
- Utilizar la alabanza con moderación. Puede distraer al estudiante del objetivo.
- Evitar la entrega de feedback de manera oral.
- Evitar entregar la respuesta correcta de forma explícita. No abusar del aspecto facilitador.
- Minimizar el uso de extensos análisis de errores y diagnósticos. El costo de producción no provee mayor beneficio para el estudiante, solo ayudan en algunas circunstancias.

Gracias