

# Determinando las dificultades en el aprendizaje de la primera asignatura de programación en estudiantes de ingeniería civil informática

**Roberto Muñoz**

Escuela de Ingeniería Civil  
Informática  
Facultad de Ingeniería, Universidad  
de Valparaíso  
Valparaíso, Chile  
roberto.munoz@uv.cl

**Marta Barria**

Escuela de Ingeniería Civil  
Informática  
Facultad de Ingeniería, Universidad  
de Valparaíso  
Valparaíso, Chile  
marta.barria@uv.cl

**René Noël**

Escuela de Ingeniería Civil  
Informática  
Facultad de Ingeniería, Universidad  
de Valparaíso  
Valparaíso, Chile  
rene.noel@uv.cl

**Eliana Providel**

Escuela de Ingeniería Civil Informática  
Facultad de Ingeniería, Universidad de Valparaíso  
Valparaíso, Chile  
eliana.providel@uv.cl

**Patricio Quiroz**

Escuela de Ingeniería Civil Informática  
Facultad de Ingeniería, Universidad de Valparaíso  
Valparaíso, Chile  
patricio.quirozv@alumnos.uv.cl

## ABSTRACT

Teaching the Programming Fundamentals is an essential part of a curriculum in Computer Science, but for most of the careers becomes a problematic aspect. Apply basic concepts and design relatively simple algorithms, appear to be somewhat difficult for the student. Based on the observations of various authors, this may be due to various factors such as motivation, learning styles, experience, etc. To determine the difficulties of the students were applied two instruments: the first in order to determine the learning styles of the students who entered in 2012 and the second one to determine which were the topics that caused more problems during the process and gets their preferences in the study. The results show that students have preferences for the visual materials, exercise guides and example programs, which is proper of the learning styles of them.

## KEYWORDS

Learning Styles, Teaching Styles, Programing Fundamentals.

## RESUMEN

La enseñanza de los Fundamentos de Programación es una parte esencial de un curriculum en Ciencias de la Computación, sin embargo para la mayoría de las carreras se transforma en un aspecto problemático. El aplicar conceptos básicos o diseñar algoritmos relativamente simples parece ser algo difícil para el estudiante. De acuerdo a lo expuesto por diversos autores, esto se puede deber a diversos factores, tales como motivación, estilos de aprendizaje diferentes, experiencia previa, entre otros. Para determinar las dificultades nuestros estudiantes se aplicaron dos instrumentos: el primero con el fin de determinar los estilos de aprendizajes de los alumnos año de ingreso 2012 y el segundo para determinar cuales

fueron los tópicos que les provocaron más problemas durante el proceso y obtener sus preferencias en el estudio. Los resultados muestran que aquellos estudiantes poseen preferencias por el material visual, guías de ejercicios y programas de ejemplo, algo que es propio al estilo de aprendizajes de ellos.

## PALABRAS CLAVE

Estilos de Aprendizaje, Estilos de Enseñanza, Fundamentos de Programación.

## INTRODUCCIÓN

En la actualidad la enseñanza de la programación es un tema de suma importancia en las carreras de Ingeniería ligadas a las tecnologías [1]. En los últimos años se han propuesto muchos enfoques y herramientas distintas, sin embargo a la fecha no parece existir un enfoque o una solución completamente satisfactoria [2]. La aplicación de conceptos básicos o el diseño de algoritmos que son relativamente simples para los docentes, parece ser algo difícil para el estudiante [3]. Estas dificultades se manifiestan independiente del paradigma y/o lenguaje utilizado. Esto se puede deber a diversos factores, tales como motivación, estilos de aprendizajes diferentes, experiencia previa, entre otros [4], [5], [6].

La deserción, cercana al 47% que se produce en los primeros años de ingeniería, provocan necesariamente mirar en forma analítica este fenómeno [7]. La carrera de Ingeniería Civil en Informática de la Universidad de Valparaíso, impartida desde el año 2005, no es ajena a esta situación. En ella se han realizado esfuerzos constantes para mejorar dicha problemática, como por ejemplo incorporación de nuevas herramientas [8], actualización de planes de estudio o nivelación de alumnos de primer año. De acuerdo a un estudio realizado por la misma carrera el año 2009, en promedio los alumnos inscribían 5,14 asignaturas por



semestre, aprobando en promedio tan sólo 1,98 asignaturas, siendo aquellas con mayor reprobación las correspondientes a Álgebra Elemental (19,5%), Fundamentos de Programación (32,4%) y Cálculo I (32,7%). Por tal razón, en una primera instancia se debe identificar, analizar y estudiar los factores que dificultan el proceso.

Los estudiantes, aprenden en forma más eficaz cuando se les enseña en su estilo personal de aprendizaje [9]. Aunque la mayoría de las personas sin discapacidad pueden aprender utilizando cualquiera de estos estilos, la mayor parte de las personas tienen más afinidad con alguno de ellos [9].

Para determinar los estilos predominantes en los estudiantes año de ingreso 2012 y los principales aspectos que dificultaron el aprendizaje de la programación, es que utilizamos los instrumentos propuestos en [10] y [11]. Éstos fueron traducidos al español y contextualizados a la cátedra de Fundamentos de Programación, que es dictada en la Escuela de Ingeniería Civil en Informática de la Universidad de Valparaíso.

## MARCO TEÓRICO

El concepto de estilo de aprendizaje no es común para todos los autores y es definido de forma variada en diversas investigaciones.

Keefe en [12] expone que los estilos de aprendizajes son *“los rasgos cognitivos, afectivos, afectivos y fisiológicos, que sirve como indicadores relativamente estables, de cómo perciben los discentes, interaccionan y responden a sus ambientes de aprendizaje”*.

Por otra parte McCarty [13] dice que: *“las personas aprenden de diferente forma, estas diferencias dependen de muchos aspectos: quiénes somos, dónde estamos, cómo visualizamos y que nos demandan las personas”*.

Para finalizar de acuerdo a Felder y Silverman [5], los estilos de aprendizaje se puede definir como: *“las fuerzas y preferencias características en la forma que tienen los estudiantes para procesar información. Algunos estudiantes pueden centrarse en el manejo de datos y diferentes tipos de algoritmos, otros se sienten mejor con los modelos matemáticos y las teorías. Algunos de ellos responden fuertemente a formas visuales de información como pinturas, cuadros, diagramas y esquemas, y otros más obtienen información de forma verbal mediante escritura y las explicaciones habladas. Algunos discentes prefieren aprender activamente e interactivamente y otros funcionan mejor de manera introspectiva e individual”*.

Para esta investigación se trabajará de acuerdo a la propuesta de [5].

### Modelo de estilos de aprendizaje propuesto por Felder y Silverman.

El estilo de aprendizaje es la forma en que un individuo aprende, cada persona tiene su forma particular de aprender, lo que se ve reflejado en

sus habilidades, intereses, debilidades y fortalezas académicas.

El modelo propuesto por Felder y Silverman se podría calificar como el modelo de las cuatro categorías bipolares. Considera cuatro categorías donde cada una se extiende entre dos polos opuestos: Activo/Reflexivo, Sensitivo/Intuitivo, Visual/Verbal, Secuencial/Global. Algunas de las características de estos estilos de aprendizaje son [5]:

1. Activos (aprenden manipulando las cosas y trabajando con otros) o Reflexivos (aprenden pensando acerca de las cosas y trabajando solos).
2. Sensitivos (concretos, prácticos, orientados hacia los hechos y los procedimientos) o Intuitivos (conceptuales, innovadores, orientados hacia las teorías).
3. Visuales (prefieren la presentación visual del material tal como películas, tablas, o diagramas de flujo) o Verbales (prefieren las explicaciones escritas o habladas).
4. Secuenciales (aprenden poco a poco en forma ordenada) o Globales (aprenden de forma holística).

## JUSTIFICACIÓN DEL ESTUDIO

Tal como se menciona en [14], es de suma importancia detectar cuales son los contenidos y/o conceptos que de acuerdo a la percepción de los estudiantes son complicados de asimilar y aplicar. De esta forma se podría implementar un sistema enseñanza alternativo con el fin de ayudar en el proceso de aprendizaje de la programación en los primeros años de las carreras de Ingeniería.

## DISEÑO DEL INSTRUMENTO.

El diseño del instrumento está basado en el cuestionario propuesto originalmente por [11] con elementos de la propuesta de [15]. Adicionalmente se ha incorporado el conocimiento acerca de los estilos de aprendizaje que presentan los estudiantes de la asignatura.

El primer cuestionario, aplicado a estudiantes que aprobaron la asignatura de programación, está compuesto por 5 secciones: 1) Perfil del Alumno (*Tipo de colegio de procedencia, año de ingreso, y oportunidad en que aprobó la asignatura*). 2) ¿Qué tarea siente que le resultó difícil aprender en la cátedra? (*ej: Usar herramientas, aprender la sintaxis, encontrar y corregir los errores de sus programas, etc.*). 3) ¿Qué conceptos de programación fueron más difíciles de aprender? (*ej. Tipos de variables, Estructuras de Repetición, E/S de datos, etc.*). 4) ¿Cuándo cree que aprende programación (*ej. Cuando lee libros, en clases teóricas, cuando practica solo, etc.*). 5) ¿Qué material considera de ayuda para aprender programación? (*ej: Libros, programas de ejemplo, guías de ejercicio, etc.*).

El segundo cuestionario tuvo como objetivo obtener una retroalimentación acerca de los estilos de aprendizaje de los estudiantes, de acuerdo a lo propuesto por [5] y disponible en [10]. Este cuestionario se utiliza para evaluar las preferencias de aprendizaje en 4 dimensiones (*activo/reflexivo, sensitivo/intuitivo, visual/verbal y global/secuencial*). Los resultados de la aplicación de ambos cuestionarios son presentados a continuación.



## RESULTADOS

Se aplicó la encuesta propuesta por [10] a 44 estudiantes de la carrera de Ingeniería Civil en Informática de la Universidad de Valparaíso. Los resultados son presentados en la Tabla 1:

	CAT 1		CAT 2		CAT 3		CAT 4	
	ACT	REF	SEN	INT	VIS	VER	SEC	GLO
Cont	28	16	32	12	34	10	29	15
%	64%	36%	73%	27%	77%	23%	66%	34%

Los resultados muestran que nuestros estudiantes son en su mayoría *Activos* llegando al 64%, por lo que el material y la docencia realizada debiese privilegiar trabajos en equipos donde abunde la discusión. El sólo tomar apuntes es complejo para ambos grupos, sin embargo es particularmente más difícil para los estudiantes activos. Tampoco se puede olvidar al 36% que corresponde a estudiantes *Reflexivos*, puesto que ellos tienen preferencias por las situaciones en que el puedan reflexionar acerca de la solución antes de actuar. Esto debido a que los estudiantes reflexivos consideran todas las opciones ante de tomar una decisión.

Por otra parte los estudiantes encuestados son en su mayoría *Sensitivos* (73%) por lo que se debiese privilegiar el trabajo en laboratorios, esto debido a que este tipo de estudiantes prefieren resolver problemas por métodos establecidos y son pacientes con los detalles. Estos estudiantes entienden mejor la información si pueden ver como se conecta con el mundo real. Las clases cuyo material sea principalmente teórico podría provocar problemas, por lo que debiese existir un equilibrio entre la teoría y su aplicación práctica. Con respecto al 27% de nuestros estudiantes que tiene más desarrollado el estilo *Intuitivo* tiende a sentirse cómodo con las abstracciones. Estos alumnos no se sienten cómodos en las clases que en su mayoría son de memorización, por el contrario se sienten extremadamente cómodos en aquellas que prima la aplicación.

De acuerdo a los resultados de la encuesta la mayor parte de nuestros alumnos es *Visual* (77%), por lo que el material que sirva de apoyo, tales como diagramas, esquemas o videos apoyará el aprendizaje de nuestros estudiantes. Para aquellos que son *Verbales*, es útil el trabajo en grupo, esto debido a que se gana comprensión de las actividades/material explicando y oyendo explicar a sus compañeros .

Para finalizar el 66% de nuestros encuestados tienen preferencia por el estilo *Secuencial*. Ellos tienden a aprender por pasos, con una secuencia que sigue lógicamente la anterior. Además a seguir caminos lógicos encontrando soluciones. Para estos estudiantes es cómodo el enfoque de enseñanza tradicional, puesto que la mayoría de las clases, bajo un enfoque de enseñanza tradicional, sigue un enfoque secuencial con una breve visión global [5].

Los cuestionarios fueron evaluados de manera automática con el instrumento definido en [10]. Éste califica la preferencia en cada una de las categorías en una escala de 1 a 11 en la cual un puntaje

de 1-3 corresponde a una preferencia discreta por un estilo, en la cual el estudiante no tendrá problemas en aprender en cualquiera de los enfoques. Si el puntaje del estudiante es entre 5 a 7 éste tendrá una preferencia moderada por una dimensión y aprenderá más fácilmente en un entorno que favorezca esa dimensión. Para finalizar si el puntaje obtenido por el estudiante es 9 u 11, éste posee una preferencia muy fuerte por una de la dimensiones, y es probable que tenga dificultades en un medio que no proporcione el entorno para dicha preferencia.

Los resultados agrupados, en discreta - moderada - fuerte, en cada una de las cuatro categorías, son presentados en los Figuras 1, 2, 3 y 4.

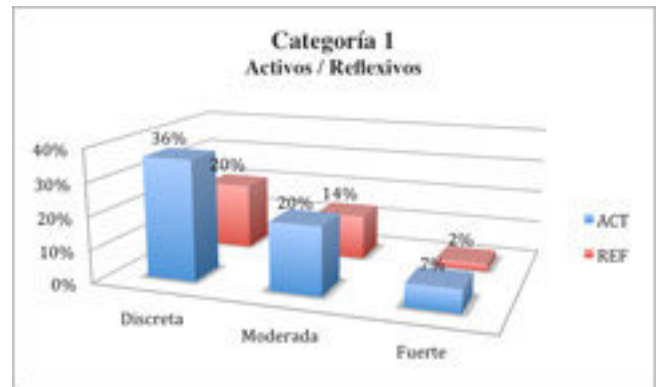


Figura 1. Categoría 1 Activos / Reflexivos.

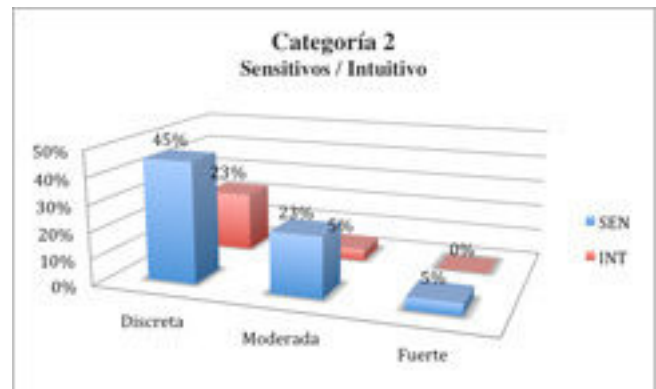


Figura 2. Categoría 2 Sensitivos / Intuitivos.

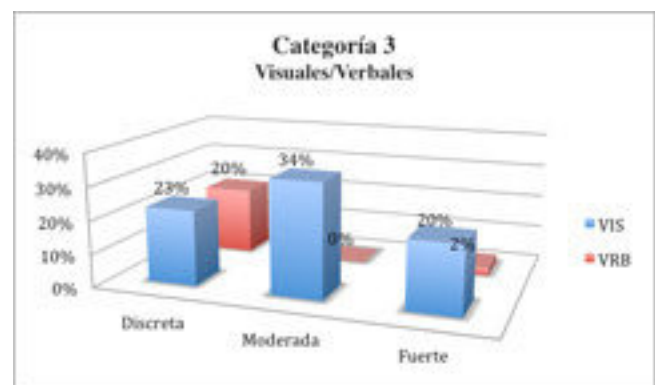


Figura 3. Categoría 3 Visuales / Verbales.



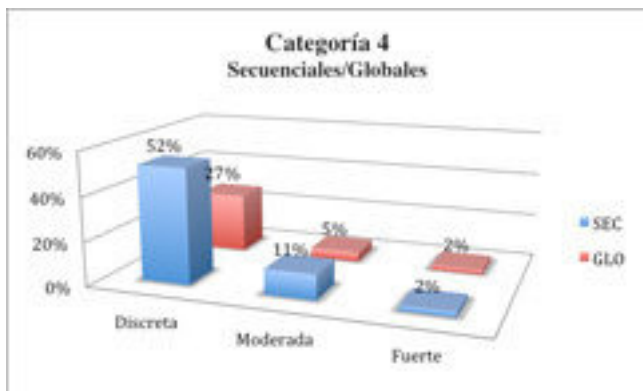


Figura 4. Categoría 4 Secuencias / Globales.

## RESULTADOS ENCUESTA APLICADA A INC102 FUNDAMENTOS DE PROGRAMACIÓN.

La segunda encuesta [16] se aplicó a 36 estudiantes que cursaron la cátedra de Fundamentos de Programación durante el primer semestre del año 2012. Los alumnos debían calificar los ítems en los puntos 2 y 3 como: Muy Difícil (MD), Difícil (D), Dificultad Media (DM), Fácil (F), Muy Fácil (MF) o No Responde (N/R). En el punto 4 como: Muy Desacuerdo (MD), Desacuerdo (D), Medianamente De Acuerdo (MeA), De Acuerdo (A), Muy de Acuerdo (MA) o No Responde (N/R), para finalizar el punto 5 debía ser evaluado como: Muy Inútil (MI), Poco Útil (PU), Medianamente Útil (MedU), Útil (U), Muy Útil (MU) y No Responde (N/R).

La participación se distribuyó de la siguiente forma.

### 1) Perfil del Alumno

El 22% de los encuestados provenía de colegios municipales, el 72% colegios Particulares Subvencionados y tan sólo el 6% de colegios Particulares pagados.

El 75% de los encuestados aprobó la cátedra cuando la cursó y el 25 % reprobó en primera oportunidad.

### 2) ¿Qué tarea siente que lo resultó difícil aprender en la cátedra? Los ítems evaluados en este punto fueron:

- Usar entornos de desarrollo (PSeInt – SLE2).
- Aprendizaje de la sintaxis del lenguaje o pseudolenguaje.
- Diseñar un pseudocódigo para resolver determinada tarea.
- Realizar ruteo manualmente.
- Convertir diagrama de flujo en pseudocódigo.
- Convertir el pseudocódigo a diagrama de flujo.
- Encontrar y corregir los errores de su propio programa.

Los resultados del punto 2) son presentados en la Tabla 2.

Preg. – Porc.	MD	D	DM	F	MF	N/R
A	6%	19%	33%	33%	8%	0%
B	3%	3%	17%	44%	33%	0%
C	8%	14%	31%	36%	11%	0%
D	3%	14%	25%	53%	6%	0%
E	3%	11%	19%	28%	36%	3%
F	6%	6%	28%	25%	36%	0%
G	8%	17%	56%	11%	8%	0%

De lo anterior se puede desprender lo siguiente:

- La mayor parte de los estudiantes no presentan dificultades al momento de aprender a utilizar un entorno de desarrollo.
- La sintaxis del pseudocódigo no es algo que complique a la mayor parte de los estudiantes, sin embargo existe un porcentaje de ellos (9%) que lo encontró difícil o muy difícil.
- El diseño de un pseudocódigo, para resolver una determinada tarea, fue una actividad que se comportó de manera similar a la anterior, sin embargo el porcentaje de estudiantes que lo encontró fácil fue superior al 30%.
- El 25 % de los encuestados encontró que el ruteo de manera manual es algo que no les traía complicaciones, algo que debiese ser natural debido a que para una gran parte de los encuestados no consideraban la sintaxis algo complicado.
- Sólo el 3 % encontró que convertir un diagrama de flujo a pseudocódigo era algo extremadamente difícil, la mayor parte de las respuesta se distribuyeron entre dificultad media y fácil.
- Las respuestas se distribuyeron de manera similar a la de E, distribuyéndose el 12 % en muy difícil y difícil, y el 53 % en dificultad media o fácil.
- Con respecto a encontrar y corregir sus propios errores, el 73 % de los encuestados respondió que era algo difícil o de dificultad media. Sólo el 8 % encontró que era muy fácil detectarlos.

### 3) ¿Qué conceptos de programación fueron más difíciles de aprender?

Los ítems evaluados en este punto fueron:

- Los tipos de datos / variables.
- Estructuras de selección.
- Estructuras de repetición.
- Arreglos unidimensionales.
- Arreglos multidimensionales.
- Algoritmos de búsqueda y ordenamiento.
- Registros.
- Archivos.
- Entrada y salida de datos.

Los resultados son presentados en la Tabla 3.

Preg. – Porc.	MD	D	DM	F	MF	N/R
A	6%	3%	14%	31%	47%	0%
B	6%	6%	19%	28%	42%	0%
C	6%	0%	31%	33%	28%	3%
D	6%	6%	17%	19%	53%	0%
E	3%	14%	44%	25%	14%	0%
F	8%	14%	31%	25%	22%	0%
G	8%	3%	28%	31%	31%	0%
H	19%	22%	28%	25%	6%	0%
I	6%	14%	31%	25%	25%	0%

De lo anterior se puede desprender lo siguiente:

- El 78% de los encuestados encontró que fue fácil o muy fácil entender el concepto de tipos de datos y variables. El 14% encontró que fue de una dificultad media y el 9% consideró que fue un aspecto difícil o muy difícil de aprender.

- B. Con respecto a estructuras de selección, tampoco se puede considerar un aspecto que los estudiantes consideren complejo de asimilar, esto debido a que el 12% de los encuestados lo considera un aspecto difícil o muy difícil de aprender.
- C. En relación a las estructuras de repetición, el 31% de los encuestados consideró que tuvo una dificultad media al momento de aprender a utilizarlas. Sin embargo sólo el 6% de los encuestados consideró que fue un tópico complicado o muy difícil de asimilar.
- D. En la pregunta de que tan complejo fue aprender arreglos unidimensionales, los estudiantes no consideraron que fuese un aspecto muy complicado, debido a que el 72% de los encuestados consideró que fue muy fácil o fácil de aprender.
- E. En lo que respecta a la pregunta de arreglos multidimensionales hubo un cambio sustancial con respecto a la pregunta anterior, puesto que el 44% de los encuestados consideró que la dificultad era media y tan sólo el 14% que era un aspecto muy fácil.
- F. Para la pregunta de la complejidad que encontraban con respecto al aprendizaje de algoritmos de búsqueda y ordenamiento el tan sólo el 22% encontró que esto era un tópico difícil o muy difícil, el resto de los estudiantes lo encontró con una dificultad media, fácil o muy fácil.
- G. Con respecto a la dificultad con respecto a registros los resultados fueron mejores que los obtenidos en los puntos E y F puesto que el 62% lo encontró un tema fácil o muy fácil de aprender.
- H. A diferencia del punto anterior, el concepto de archivos fue el que provocó mayor complicación, puesto que el 69% de los encuestados consideró que fue un aspecto muy difícil, difícil o de dificultad media de asimilar y aprender.
- I. Entrada y salida de datos no es un aspecto complicado para los estudiantes, los resultados muestran que el 50% de los encuestados consideró que fue un tópico fácil o muy fácil de aprender

4) *¿Cuándo crees que aprendes programación?*

Los ítems evaluados en este punto fueron:

- A. Cuando lee libros de programación.
- B. Cuando lee apuntes de clases.
- C. En las clases teóricas.
- D. En las clases prácticas.
- E. Cuando estudia/practica solo.
- F. Cuando estudia/practica en grupo.
- G. Cuando practica en papel.
- H. Cuando practica en el computador.

Los resultados son presentados en la Tabla 4.

Preg. - Porc.	MD	D	MeA	A	MA	N/R
A	14%	8%	33%	25%	14%	6%
B	8%	8%	25%	31%	19%	8%
C	3%	6%	28%	42%	17%	0%
D	6%	3%	11%	36%	42%	3%
E	8%	11%	17%	42%	19%	3%
F	6%	8%	33%	17%	25%	11%
G	0%	3%	28%	33%	31%	6%
H	3%	3%	28%	28%	33%	6%

lo anterior se puede desprender lo siguiente:

- A. El 33 % está medianamente de acuerdo en que aprende con libros de programación. El 25 % está de acuerdo con que aprendía en este tipo de elementos. Por otra parte el 22 % considera que no aprende en este tipo de elementos
- B. En lo que respecta a si los estudiantes consideran los apuntes de clases son útiles para su aprendizaje, el 50 % de los encuestados está de acuerdo o muy de acuerdo con esta afirmación, el 25 % estaba medianamente de acuerdo y sólo el 16 % no está de acuerdo muy en desacuerdo.
- C. Con respecto a la afirmación si consideran que aprenden en las clases de teoría, el 28 % está medianamente de acuerdo con esta afirmación, el 42 % está de acuerdo con la afirmación, tan sólo el 9 % está en desacuerdo o muy en desacuerdo.
- D. En relación a las clases prácticas el 78 % está muy de acuerdo o desacuerdo. Esto claramente corroborado por el primer instrumento utilizado en la que la mayor parte de los estudiantes eran activos. Tan sólo el 14 % no está de acuerdo con esta afirmación.
- E. El 61% de los estudiantes considera que aprende más cuando estudia programación. Por otra parte tan sólo el 19% no está de acuerdo con esta afirmación.
- F. El 42 % está de acuerdo o muy de acuerdo con que aprende cuando estudia o practica programación en un grupo, el 33 % está medianamente de acuerdo con esta afirmación y el 14 % no está de acuerdo.
- G. El 90 % de los encuestados estuvo desde medianamente de acuerdo a muy de acuerdo. en que aprendía al practicar utilizando papel.
- H. Para finalizar los estudiantes también consideran que aprenden más cuando practican en computador, puesto que el resultado es similar al anterior, por lo que la enseñanza de la primera asignatura debe poseer un equilibrio entre estos últimos 2 elementos.

4) *¿Qué material considera que es de ayuda para aprender a programar?*

Los ítems evaluados en este punto fueron:

- A. Libros de programación.
- B. Apuntes de cátedra.
- C. Guías de ejercicio.
- D. Programas de ejemplo.
- E. Internet.

Los resultados son presentados en la Tabla 5.

Preg. - Porc.	MI	PU	MedU	U	MU	N/R
A	3%	11%	22%	42%	22%	0%
B	6%	3%	14%	36%	42%	0%
C	3%	3%	14%	28%	53%	0%
D	3%	0%	14%	19%	64%	0%
E	6%	17%	36%	31%	11%	0%

De lo anterior se puede desprender lo siguiente:

- A. El 64 % de los encuestados considera que los libros son un elemento útil o muy útil, sólo el 14 % no los considera una ayuda.



B. El 78% de los estudiantes consideran más útiles los apuntes de cátedra, que los libros. Tan sólo el 9 % no encontró que tenían utilidad.

C. Las guías de ejercicio también fueron un elemento que los estudiantes consideraron extremadamente útiles superando el 81% aprobación, sólo el 3% no las considera de utilidad.

D. Con respecto a los programas de ejemplo el 83 % consideró que son un elemento útil y muy útil, solo el 3% no lo consideró de utilidad.

El 78% considera que Internet (en lo que respecta a información complementaria para entender la asignatura) es un elemento que ayuda a aprender programación..

Además, para detectar algún tipo de relación que no fuese visible de manera simple, utilizamos el software de minería de datos Weka [17] y ejecutamos el algoritmo de nombre *a priori*. Éste permite determinar reglas de asociación en un conjunto de datos.

Con respecto a estas respuestas, se puede presentar la regla que más nos llamó la atención:

Una de las reglas interesantes identificadas, fue que un 75% de los alumnos que consideraban el Ruteo y los Algoritmos de Búsqueda eran materias de dificultad media o compleja, finalmente reprobó el curso.

De acuerdo a lo presentado en el párrafo anterior, al momento de detectar problemas en este tipo de tópicos sería necesario y/o recomendable aplicar mecanismos de reforzamiento acordes a su estilo de aprendizaje.

## CONCLUSIONES

En este trabajo fueron presentado los resultados de las encuestas respondidas por estudiantes de Ingeniería Civil en Informática de la Universidad de Valparaíso, con el fin de determinar cuales son los aspectos que consideran más complejos al momento de aprender programación, además de presentar los resultados que tienen relación con sus preferencias y estilos de aprendizaje.

Al momento de realizar docencia se debe considerar los estilos de aprendizaje de los estudiantes, puesto que este es un factor de suma importancia, no en el impacto en cuanto aprenden las personas, sino que más bien en su satisfacción con el proceso [18].

Se debe tomar en consideración no sólo los estilos de aprendizaje predominantes de los estudiantes, sino también los propios, para de esta forma no favorecer el potencial de aprendizaje y actitud de los estudiantes que comparten los mismos estilos o afectar a los que tienen diferentes estilos de aprendizaje de los nuestros. Las diferencias de estilos de aprendizaje predominantes se puede ver como una oportunidad de desarrollo docente [19]. Como

trabajo futuro se espera aplicar estos instrumentos a distintas cátedras, además de también aplicarlos a los docentes de estas asignaturas, de esta forma contrastar las percepciones en lo que respecta a la complejidad de los contenidos de los estudiantes con sus profesores. Además se espera aplicar técnicas de minería de datos para poder encontrar relaciones entre los estilos de aprendizaje y tópicos que influyen en la reprobación de los estudiantes.

## REFERENCIAS

1. Blake, J.D. Language considerations in the first year CS curriculum. *J. Comput. Sci. Coll.* 26, 6 (2011), 124–129.
2. Bozorgmanesh, M., Sadighi, M., and Nazarpour, M. Increase the efficiency of adult education with the proper use of learning styles. *Nature and Science* 9, 5 (2011).
3. Carbone, A., Hurst, J., Mitchell, I., and Gunstone, D. An exploration of internal factors influencing student learning of programming. *Proceedings of the Eleventh Australasian Conference on Computing Education - Volume 95*, Australian Computer Society, Inc. (2009), 25–34.
4. Casas, S. and Vanoli, V. Programación y Algoritmos: Análisis y Evaluación de Cursos Introductorios. (2007).
5. Felder, R.M. and Brent, R. Understanding Student Differences. *Journal of Engineering Education* 94, 1 (2005), 57–72.
6. Felder, R.M. and Silverman, L.K. Learning and Teaching Styles in Engineering Education. *Engineering Education* 78, 7 (1988), 674–681.
7. Felder, R.M. and Soloman, B.A. Learning Styles Questionnaire. <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>. <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>.
8. González, L., Uribe, D., and González, S. *Estudio sobre la repitencia y deserción en la educación superior chilena*. IESALC - UNESCO, 2005.
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The WEKA data mining software. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10.
10. Hannum, W. Training myths: False beliefs that limit training efficiency and effectiveness, part 2. *Performance Improvement* 48, 6 (2009), 25–29.
11. Hernández, L. La importancia de los estilos de aprendizaje en la enseñanza de inglés como lengua extranjera. *Revista Espéculo (Universidad Complutense)* 27, IX (2004).
12. Keefe, J.W. *Learning style: theory and practice*. National Association of Secondary School Principals, 1987.
13. Lahtinen, E., Ala-Mutka, K., and Järvinen, H.-M. A study of the difficulties of novice programmers. *ACM SIGCSE*

*Bulletin 37, 3 (2005), 14.*

14. McCarthy, B. and McCarthy, D. Teaching Around the 4MAT® Cycle: Designing Instruction for Diverse Learners with Diverse Learning Styles. SAGE Publications, 2005.

15. Mendes, A.J. and Marcelino, M.J. Tools to support initial programming learning. (2006).

16. Muñoz, R., Barría, M., and Rusu, C. Mundos Virtuales Como Apoyo a la Docencia en Ingeniería. .

17. Muñoz, R., Pérez, F., and Meza, J. Encuesta realizada a los alumnos de cursaron INC102-Fundamentos de Programación.

<https://docs.google.com/a/uv.cl/spreadsheet/viewform?formkey=dFNhaWtGaWFGZnZ4bG5BaUI0M253d0E6MQ#gid=0>, 2012.

18. Tan, P.-H., Ting, C.-Y., and Ling, S.-W. Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception. *Computer Technology and Development*, 2009. ICCTD '09., IEEE (2009), 42–46.

19. Villalobos, J., Casallas, R., and Marcos, K. El Reto de Diseñar un Primer Curso de Programación de Computadores. (2005).

