

Análisis de la evolución de un curso: productividad y desigualdad *

Maria J. Blesa Amalia Duch Joaquim Gabarró Jordi Petit Maria Serna
Departament de Ciències de la Computació
Universitat Politècnica de Catalunya — BarcelonaTech, 08034 Barcelona
{mjblesa, duch, gabarro, jpetit, mjserna}@cs.upc.edu

Resumen

Este trabajo analiza cuantitativamente las diferentes medidas adoptadas en la evaluación continua del curso de Programación-1 de la Facultat d'Informàtica de Barcelona de la Universitat Politècnica de Catalunya desde 2010, año en que comienza la impartición del nuevo grado en Ingeniería Informática. Se analiza el efecto de dichas medidas sobre las categorías de calificaciones obtenidas por los estudiantes en relación al incremento de carga docente del profesorado. El estudio de la evolución de estos dos valores a lo largo del tiempo usa una aproximación de coste-beneficio marginal, clásica en ámbitos de economía. Usando herramientas provenientes de las ciencias sociales, este estudio se complementa con un análisis preliminar de desigualdad. Ambos enfoques pretenden introducir nuevas técnicas para el análisis objetivo del impacto de medidas docentes.

Abstract

This paper analyzes quantitatively the different measures taken in the continuous assessment of the CS1 course at the Facultat d'Informàtica de Barcelona of the Universitat Politècnica de Catalunya since 2010, when the new degree in Informatics Engineering began. The effect of those measures on the marks obtained by the students is analyzed with respect to the increment of the course load for the faculty members. These two values are compared along time through a classical marginal cost-benefit approach. Using tools commonly accepted in the social sciences, this study is complemented with a preliminary analysis of inequality. Both approaches aim at introducing new techniques for an objective analysis on the impact of teaching measures.

*Trabajo parcialmente financiado por la mención de la Generalitat de Catalunya a ALBCOM como grupo de investigación consolidado (ref. SGR 2014:1034), por los fondos FEDER de la Unión Europea, y por los proyectos TIN2013-46181-C2-1-R y TIN2012-37930-C02-02 del Ministerio de Economía y Competitividad.

Palabras clave

Programación-1, evaluación continua, productividad, desigualdad, coeficiente de Gini.

1. Introducción

Programación-1 (PRO1) es la primera asignatura de programación del Grado de Ingeniería Informática en la Facultat d'Informàtica de Barcelona (FIB) de la Universitat Politècnica de Catalunya (UPC). Desde septiembre de 2006 el curso sigue una metodología "learning-by-doing" [1, 2, 3, 16] (aprendizaje mediante práctica) basada en una colección estratégicamente seleccionada y organizada de ejercicios de programación que deben ser resueltos usando C++ con un juez de programación. Desafortunadamente, los dos primeros años de implantación no dieron los resultados esperados en la asignatura [8].

Por este motivo, el profesorado de PRO1 ha introducido, paulatinamente, una serie de medidas con el propósito de motivar a los estudiantes a trabajar más, mejor, y de manera más autónoma y continua para así adquirir los conocimientos teóricos y prácticos requeridos para superar la asignatura. Todo ello sin sacrificar los objetivos generales, el nivel requerido y el enfoque del curso. En consecuencia, el curso ha experimentado diversas modificaciones que han incrementado simultáneamente el trabajo de evaluación continua de los estudiantes y la carga docente del profesorado de la asignatura.

Con el fin de analizar objetivamente las diferentes medidas ya introducidas en la asignatura y, sobre todo, de poder predecir el efecto de nuevas medidas antes de su introducción, en [7] se definió una medida de la productividad en el aprendizaje. Con esta medida se correlacionan los resultados académicos obtenidos por los estudiantes con el tiempo dedicado por los profesores de la asignatura.

El análisis de la evolución temporal de las actividades evaluadoras de un curso es un tema de estu-

dio consolidado [12, 17]. El enfoque aplicado al curso de PRO1 que se propone en [7] se basa en la aproximación de coste-beneficio marginal clásica en economía [4, 10, 21], y permite relacionar el número de estudiantes que aprueban con la carga docente mediante una curva de productividad, de tal forma que es posible evaluar el impacto de las medidas pedagógicas tomadas.

En este trabajo refinamos el análisis realizado en [7] en relación a la evolución de PRO1, desde el curso 2010–2011, en dos direcciones, fundamentadas en criterios de desagregación. En primer lugar desagregamos las calificaciones de acuerdo con los criterios habituales de nuestro sistema de calificaciones (Sobresaliente, Notable, etc.). Basándonos en la metodología propuesta en [7], aplicamos un análisis coste-beneficio por rangos de notas. En segundo lugar, desagregamos las calificaciones en base a un criterio utilizado en ciencias sociales para analizar la desigualdad y que se basa en las *tablas sociales* y la *aritmética política* del siglo XVIII [6]. En este trabajo hemos optado por desagregar las calificaciones de PRO1 aplicando una distribución (10%, 40%, 50%), que es habitual en sociología (p. ej. en [14, cap. 7]). Con esta distribución se desagregan las calificaciones en segmentos para analizar el 10% de la población con mayor nota (*clase alta*), luego el 40% que sigue (*clase media*) y finalmente el 50% restante (*clase baja*). Para cada una de estas poblaciones proponemos un análisis basado en la nota media por segmento de población, combinada con una medida de la desigualdad social (que se mide habitualmente mediante el *coeficiente de Gini* [5, 9]).

La complejidad de un curso como PRO1 no puede capturarse en su totalidad mediante un análisis cuantitativo a partir de los datos de que disponemos. Sin embargo, los autores sí creemos que los resultados obtenidos y la metodología propuesta pueden ser de gran utilidad al considerar la introducción de mejoras tendentes a conseguir un mayor número de estudiantes que alcanza el nivel requerido en la asignatura de programación.

El contenido de este artículo se estructura de la siguiente manera: en la Sección 2 se presenta el contexto de la asignatura y las medidas docentes que ha ido adoptando en su evolución de los últimos cinco cursos, así como los conceptos en los cuales basaremos nuestro análisis posterior. La Sección 3 realiza un análisis coste-beneficio para estudiar la productividad partiendo de una desagregación clásica por rangos de las calificaciones de los estudiantes. En cambio, la Sección 4 presenta un análisis alternativo basado en una desagregación por segmentos porcentuales de la población y en las desigualdades que esta presenta. Finalmente, las conclusiones y líneas de investigación futura de este trabajo se detallan en la Sección 5.

2. Contexto

La FIB ofrece el curso de PRO1 dos veces en cada año académico: en el semestre de otoño (S1) y en el de primavera (S2). En S1 llegan todos los estudiantes de nueva admisión y, para un gran número de ellos, PRO1 es su primer curso de programación. En S2 los estudiantes son mayoritariamente repetidores. Contrariamente a lo que parecería lógico, el hecho de tener algunos conocimientos sobre la materia de PRO1 parece condicionar un alto nivel de absentismo en las clases del S2. A pesar de la diferente naturaleza de los estudiantes de cada semestre, ni la organización del curso ni la forma de evaluación cambian entre S1 y S2. Lo que sí se suele adaptar a la audiencia concreta de cada semestre es el contenido de las clases, especialmente el de las clases prácticas.

El curso de PRO1 se compone de 2 horas de clases de teoría y 3 horas de clases de práctica a la semana. En ambos semestres, las clases de teoría se organizan en grupos de 60–80 estudiantes y las de práctica en grupos de 15–20.

El objetivo principal de PRO1 siempre ha sido asegurar que los estudiantes aprenden y dominan habilidades básicas de programación. A partir del curso 2006–2007 se hizo una apuesta por alcanzar dicho objetivo con un claro refuerzo de la parte práctica de la asignatura. Por eso el curso se (re)organizó alrededor de concepto de “problemas de programación”, es decir, ejercicios de programación descritos con enunciados muy claros en términos de entradas válidas y salidas correctas. Para cada ejercicio, los estudiantes han de escribir un programa en C++ que lo resuelva y se comporte acorde con el enunciado. Para aplicar esta metodología de aprendizaje mediante práctica, se hace uso del *juez de programación* Judge.org [13]. El juez es una herramienta en línea que, automáticamente y en tiempo real, verifica si el programa propuesto por un estudiante es una solución correcta a un ejercicio de programación. Dicha herramienta de soporte está disponible para los estudiantes las 24 horas del día, proporciona una amplia fuente de ejercicios y fomenta la autoorganización del proceso de aprendizaje. A nivel docente permite dinamizar enormemente las clases prácticas y hacer un fácil seguimiento del trabajo de los estudiantes, así como diseñar, organizar y administrar exámenes. Este tipo de jueces surgieron de concursos de programación (p. ej., el UVa Online Judge [15]) y su uso ha sido adaptado en los últimos tiempos a entornos educativos [11, 13, 20, 22].

La primera colección de ejercicios se diseñó convenientemente en 2006 y se organizó por temas y dificultad. El tamaño actual de la colección, con más de 300 problemas, permite tener material suficiente tanto para la selección de los contenidos de las sesiones de laboratorio, como para ejercicios complementarios. Du-

rante el curso se espera que los estudiantes resuelvan el máximo número de problemas posible. En el curso 2006–2007, los exámenes también se adaptaron a este formato y se conforman de problemas de dificultad similar a los de la colección.

Evolución. A lo largo de los cursos se han ido tomando una serie de medidas para intentar incrementar el número de aprobados sin merma de contenidos ni conocimientos requeridos. Siguiendo la notación introducida en [7], hacemos referencia a cada una de las medidas introducidas desde el curso 2010–2011 al 2014–2015 como t_4, \dots, t_8 , respectivamente.

t_4 **Grado en Informática (2010–2011):** Se introduce un nuevo plan de estudios, que sigue vigente actualmente. El curso se estructura en 2 horas de teoría y 3 horas de laboratorio semanales. La evaluación continuada incluye la resolución de ejercicios prácticos en forma de exámenes parciales con un peso importante (60%) en la nota final.

t_5 **Listas de problemas (2011–2012):** Para fomentar el trabajo continuado y facilitar una evaluación continua positiva, se diseñan listas de problemas adicionales para cada tema del curso. El derecho a realizar un examen parcial se condiciona a resolver como mínimo el 70% de los ejercicios de las listas correspondientes. En contrapartida los parciales se componen de ejercicios de dichas listas.

t_6 **Curso de reevaluación (2012–2013):** La FIB ofrece la posibilidad de realizar un curso intensivo y un examen adicional de reevaluación a los estudiantes suspendidos con nota igual o superior a 3.5. Dicho curso es una síntesis de 12 horas y tiene lugar una vez ha acabado el curso convencional. La asistencia a las clases es obligatoria y se mantiene el requerimiento de resolver como mínimo el 70% de los ejercicios propuestos para tener derecho a examen. Para aquellos alumnos que aprueban el examen de reevaluación la nota final es 5.0; el resto mantiene la calificación del curso.

t_7 **Exámenes parciales con nuevos ejercicios (2013–2014):** Se cambia el diseño de los exámenes parciales. Ahora pasan a componerse exclusivamente de problemas nuevos que los estudiantes no conocen de antemano. Se pretende evitar un problema de mecanización del aprendizaje.

t_8 **Diversificación del curso (2014–2015):** Hasta 2014 la colección de problemas seleccionada para PRO1 se mantuvo prácticamente invariable. Con el objetivo de diversificar los contenidos del curso y ayudar a los estudiantes a afrontar y resolver problemas diferentes, se incorporan problemas nuevos a la colección y se diseñan colecciones distintas para los semestres S1 y S2.

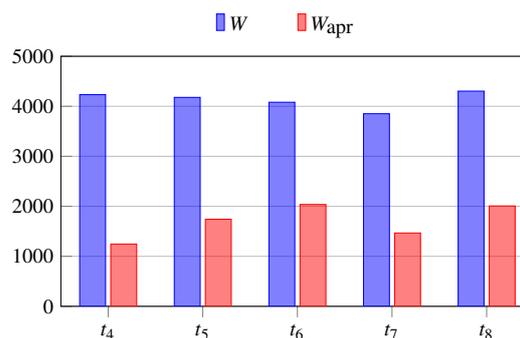


Figura 1: Trabajo total y trabajo destinado a Apr.

	Est	Apr	Sob	Not	Suf	Rev	NR	NP
t_4	514	132	11	36	85	65	243	74
t_5	492	191	31	64	96	56	198	47
t_6	465	211	44	86	81	57	127	70
t_7	436	155	11	49	95	36	152	93
t_8	448	209	24	72	113	32	101	106

Figura 2: Número de estudiantes de los cursos t_4 a t_8 por rangos de notas.

Trabajo. La Figura 1 muestra, para cada periodo, la carga estimada de trabajo docente total (W) medido en horas y, asumiendo que el trabajo por matriculado es uniforme, la estimación del trabajo dedicado a los aprobados (W_{apr}). Referimos al lector al artículo [7] para una descripción completa del desglose de la estimación del trabajo y de la metodología utilizada para su obtención. La estimación de la cantidad de trabajo se obtuvo mediante una encuesta entre el profesorado de la asignatura (aprox. 20 docentes) que incluía una valoración de las horas destinadas a cada una de las siguientes tareas: (1) Diseño, prueba y preparación de exámenes; (2) sesiones de teoría; (3) sesiones de laboratorio; (4) corrección de exámenes; (5) vigilancia de controles; (6) coordinación del curso; (7) rediseño del curso; y (8) mantenimiento del software.

Calificaciones. Por razones de espacio centramos este estudio en el análisis de las calificaciones obtenidas en el S1 de cada periodo. Como se muestra en [7], el comportamiento del S2 es muy distinto al del S1 al tratarse de alumnos que en su mayor parte son repetidores. Si embargo, los resultados para los datos agregados del S1 y S2 son bastante similares a los del S1.

3. Productividad

El objetivo de este apartado es extender el análisis coste-beneficio de [7] a los grupos correspondientes a una desagregación clásica de las notas. Queremos analizar cómo se desagrega la función de productividad y

estudiar las variaciones en cada grupo de estudiantes. Es decir, queremos determinar si el efecto (positivo o negativo) de una medida con respecto a la productividad del total de aprobados se mantiene o cambia dependiendo de la categoría de notas.

Categorías de notas. Para realizar el estudio de la productividad por grupos hemos desagregado las calificaciones (en el rango NP , $[0, 10]$) en seis rangos. Tres de ellos corresponden a los aprobados Apr ($[5, 10]$) y se corresponden con la calificación habitual: Sob , sobresaliente ($[9, 10]$); Not , notable ($[7, 9)$); Suf , suficiente ($[5, 7)$). Las notas de los no aprobados Sus se han desagregado en 3 rangos: Rev , reevaluable ($[3.5, 5)$); NR , no reevaluable ($[0, 3.5)$); NP , no presentado (NP). Entendemos que el rango de reevaluables es importante ya que algunas de las medidas tomadas pretenden incidir específicamente en este grupo. En nuestros estudios la calificación de NP se otorga cuando el estudiante no ha participado en ningún acto evaluativo con peso superior a 20%. Tales actos evaluativos tienen lugar al final del curso por lo que no podemos presuponer un abandono temprano de la asignatura. En la Figura 2 se muestra el desglose de las calificaciones para las notas del S1 según las categorías propuestas. Est representa el número total de estudiantes matriculados.

Productividad. La *productividad* Π de un curso se define en [7] como $\Pi = P/W$, donde P es el número de estudiantes aprobados (categoría Apr) y W el número de horas de dedicación docente.

En este trabajo utilizamos esta medida de productividad aplicada a cada una de las distintas poblaciones, es decir, que en vez de utilizar el total de aprobados en la fórmula anterior, utilizamos el número de estudiantes que se encuentra en cada una de las clases de estudio. Así $\Pi(C) = \#C/W$ donde $\#C$ es el número de estudiantes en la categoría C objeto de estudio. Notemos que, si $n = \#Est$ es el número de estudiantes, el trabajo docente por estudiante es $W_{ind} = W/n$. Así la productividad en la clase Est es $\Pi(Est) = \#Est/W = n/W = 1/W_{ind}$, es decir la inversa del trabajo docente dedicado a cada estudiante.

La noción de *productividad útil* refina el análisis de las diferentes categorías de notas dentro de la clase de estudiantes aprobados. La definimos como $\Pi_u(C) = \#C/W_{apr}$, siendo W_{apr} la estimación del trabajo total dedicado a los aprobados, es decir, $W_{apr} = \#Apr \cdot W_{ind}$.

Análisis de la productividad. Las Figuras 3a, 3b y 3c muestran las curvas de productividad por categoría correspondientes a las calificaciones del S1. Puede verse que la curva de productividad $\Pi(Est)$, aún con poca variabilidad, es decreciente, lo que indica un incremento continuado del trabajo docente por estudian-

te. Con respecto la productividad en los distintos grupos, cabe mencionar que los grupos Sob y Not muestran el mismo comportamiento. La tendencia es similar a la curva de productividad de Apr . Esto permite observar que las medidas adoptadas en la asignatura tienen el mismo efecto en la productividad relativa a estas dos poblaciones y que las diferencias en tendencia con la curva de productividad de Apr provienen de la productividad de la clase Suf . Como se puede observar, la gráfica de Suf no tiene la misma forma que la de Apr , y en t_6 la productividad en Suf baja mientras que no lo hace en Sob , Not y Apr . Esto indica que la medida tomada en t_6 tiene un efecto contrario en Suf que en Sob y Not .

Las gráficas de NR y Apr tienen tendencias cuasi simétricas (o complementarias) lo que parece indicar que la variación de productividad de Apr contrarresta, en cierta medida, la variación en productividad de NR .

Parece muy notorio que el efecto de todas las medidas adoptadas es contrario en la clase Apr y en la clase NR , ya que las curvas son cuasi simétricas y en ambos casos con pendientes bastante pronunciadas. Esto indicaría que una productividad creciente de aprobados se corresponde con una productividad decreciente de no reevaluables, y viceversa. Por tanto, las medidas afectan en sentido inverso a las dos categorías.

La curva de productividad de Rev es claramente decreciente, y decrece más rápidamente a partir de t_6 , lo que parece razonable al haberse introducido medidas orientadas a minimizar el número de estudiantes en esta clase.

Con respecto a la clase NP puede observarse que, excepto en el tramo de t_4 a t_5 , la productividad es creciente y, en el periodo de t_5 a t_7 , bastante pronunciada. Sin embargo, la productividad real en esta clase puede no ser visible al haber adoptado un criterio de reparto equitativo del trabajo docente con relación a los estudiantes matriculados.

Como primera conclusión vemos que el efecto en la productividad de las medidas que se han tomado no se desagrega con la misma tendencia (véanse las tendencias de la clase Apr en las clases Sob y Not). Con relación a las categorías de alumnos que suspenden, las tendencias de las clases Rev y NR son similares, con una tendencia diferente en la clase NP .

Análisis de la productividad útil. La Figura 3d muestra las curvas de productividad útil para las poblaciones Sob , Not y Suf . El resultado, reafirma la observaciones sobre la curva de productividad, con un efecto más pronunciado. La productividad en Suf parece ir en sentido contrario que en Sob y Not . Esto refleja que las horas de trabajo W_{apr} son productivas principalmente en las clases de notas superiores.

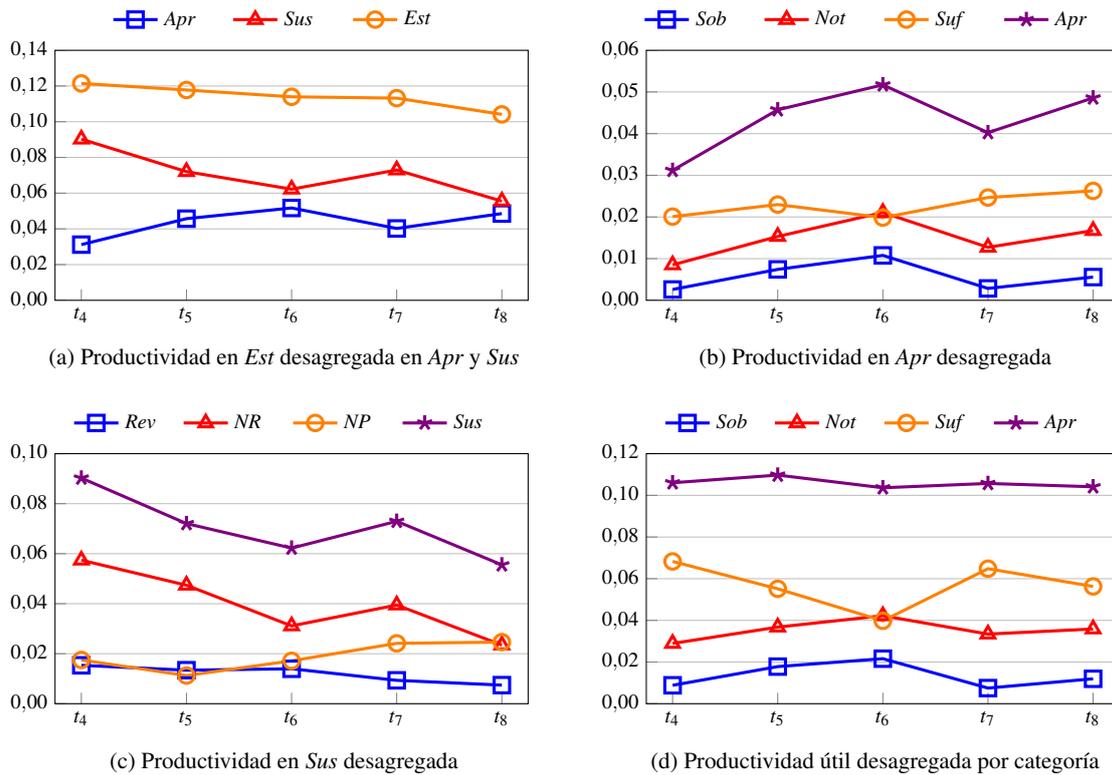


Figura 3: Desagregaciones de la productividad.

Ganancia marginal. La noción de *ganancia marginal*, definida también en [7], complementa el análisis de la productividad proporcionando una mayor exactitud en el análisis de las variaciones entre periodos consecutivos. Extendiendo el modelo a la población desagregada definimos la *variación de trabajo* como

$$\Delta W_t = W_t/N_t - W_{t-1}/N_{t-1},$$

donde W_t es el trabajo total en el período t y N_t el número total de matriculados. Definimos también la *variación en la categoría C* como

$$\Delta C_t = C_t/N_t - C_{t-1}/N_{t-1},$$

donde C_t es el número de estudiantes en la categoría C en el período t . Usando los valores mostrados en [7], se clasifican las medidas en cinco categorías, tal y como se refleja en la Figura 4, y se asocia un nivel de bondad a la medida implementada. La Figura 5 proporciona los valores numéricos de la variación para las categorías *Sob*, *Not*, *Suf* y *Apr* junto con la matriz de clasificación alfabética. Como se puede apreciar, la medida introducida en t_5 es muy positiva y lo es tanto en *Apr* como en todas sus subcategorías. Lo mismo ocurre con t_8 para una medida que se puede clasificar como positiva. El efecto de t_6 y t_7 es opuesto para la clase *Suf*. Así, t_6 es muy negativa para *Suf* y positiva en el resto mientras

	$\Delta W_t > 0$	$\Delta W_t < 0$
$\Delta C_t < 0$	(a) Muy negativa	(b) Negativa
$\Delta C_t > 0$	$\Delta W_t > \Delta C_t$ $\Delta W_t < \Delta C_t$	(c) A evitar (d) Positiva
		(e) Muy positiva

Figura 4: Clasificación de medidas de [7].

que t_7 es positiva para *Suf* y muy negativa para el resto. Se reafirma así la tendencia opuesta que se apreciaba en los análisis de productividad.

4. Desigualdad

Hasta ahora se han analizado los resultados desagregando las notas según la división académica usual. Esta sección es un primer intento de análisis desagregando las notas utilizando algunos de los métodos bien conocidos en ciencias sociales y economía. También se pretende ver hasta qué punto dichos métodos permiten arrojar nueva luz a los análisis anteriores o permiten mostrar aspectos complementarios. En consecuencia, proponemos un análisis transversal que se basa en la evolución de dos componentes: Por una parte la nota media y la nota de corte de los segmentos y por otra el *coeficiente de Gini* [5, 9].

	ΔSob	ΔNot	$\Delta Stuf$	ΔApr	ΔW
t_5	4,16	6,00	2,98	13,14	-0,09
t_6	3,16	5,49	-2,09	6,56	0,28
t_7	-6,94	-7,26	4,37	-9,83	0,06
t_8	2,83	4,83	3,43	11,10	0,77

(a) Categorías

	ΔSob	ΔNot	$\Delta Stuf$	ΔApr
t_5	e	e	e	e
t_6	d	d	a	d
t_7	a	a	d	a
t_8	d	d	d	d

(b) Trabajo

Figura 5: Variación en las categorías y en el trabajo.

Segmentos de la población y nota media. La distribución (10%, 40%, 50%) con relación a una medida de riqueza (por ejemplo la renta) consiste en desagregar la población en tres sectores [14]. El segmento alto formado por la población que corresponde al 10% con valor más alto. El segmento medio, correspondiente al 40% restante. El 50% restante de la población constituye el segmento bajo. En nuestro símil, utilizaremos las calificaciones como medida de riqueza de la población de estudiantes. En este análisis tratamos la nota NP como si fuese un cero, ya que hemos tratado a esta categoría de estudiantes como estudiantes activos a los que se les asignan recursos. Dado que los estudiantes abandonan a lo largo del curso y lo hacen de múltiples maneras sería necesario un estudio adicional para poder reevaluar el coste asociado a estos estudiantes. Usaremos los identificadores Ea , Em y Eb para referirnos a los sectores alto, medio o bajo de la población de estudiantes matriculados Est y los identificadores Aa , Am , Ab para referirnos a los sectores correspondientes de la población de estudiantes aprobados Apr . En la Figura 6 se muestran las evolución de las *notas medias* y *notas de corte* en cada uno de los sectores.

Con este tipo de desagregación podría intentarse hacer el mismo análisis de productividad que se hizo para la desagregación por notas, sin embargo, en este caso dicho análisis no es útil. Recordemos que la productividad para el segmento S es

$$\Pi(S) = \frac{1}{W_{ind}} \cdot \frac{\#S}{n}$$

Cuando S es un porcentaje fijo de la población, su productividad es la inversa del trabajo individual escalada por el porcentaje correspondiente. Así, las curvas de productividad de la Figura 3 no proporcionan información adicional. La alternativa que proponemos es estudiar la desigualdad mediante el coeficiente de Gini.

	Est Med	Ea Med	$Ea-Em$ Corte	Em Med	$Em-Eb$ Corte	Eb Med
t_4	2,83	8,24	6,71	4,42	2,01	0,50
t_5	3,73	9,12	8,10	5,79	3,40	1,02
t_6	4,39	9,69	8,92	6,70	4,28	1,48
t_7	3,49	9,06	7,90	5,36	3,10	0,89
t_8	3,80	8,95	7,90	5,90	3,80	1,12

(a) Matriculados

	Apr Med	Aa Med	$Aa-Am$ Corte	Am Med	$Am-Ab$ Corte	Ab Med
t_4	6,77	9,45	8,78	7,49	6,37	5,69
t_5	7,10	9,69	9,40	7,96	6,90	5,91
t_6	7,59	10,0	9,78	8,56	7,55	6,33
t_7	6,77	9,84	9,10	7,79	6,30	5,39
t_8	6,97	9,72	9,00	7,73	7,00	5,84

(b) Aprobados

Figura 6: Notas promedio y notas de corte para los distintos segmentos.

Coefficiente de Gini. La desigualdad de una sociedad se mide habitualmente mediante el coeficiente de Gini. Esta medida también se ha utilizado en ámbitos educativos [18, 19]. Formalmente, si $r = (r_1, \dots, r_n)$, con $n > 1$, es un vector de resultados, y su media es $\mu = \frac{1}{n} \sum_{i=1}^n r_i$, el coeficiente de Gini se define como

$$gini(r) = \frac{\sum_{i=1}^n \sum_{j=i+1}^n |r_i - r_j|}{\mu \cdot n \cdot (n-1)}$$

Cuando r representa un conjunto de notas, el coeficiente de Gini mide lo separadas que están unas de otras y, por lo tanto, mide el grado de heterogeneidad de las notas. Intuitivamente, un coeficiente de Gini próximo a 1 indica una fuerte desigualdad, mientras que uno próximo a 0 indica una distribución igualitaria. Así pues, en nuestro contexto, el coeficiente de Gini mide el grado de heterogeneidad (o desigualdad) de conocimientos de los estudiantes. Parece tentador asociar un coeficiente de Gini alto a clases con gran heterogeneidad de conocimientos. Un coeficiente alto sería un indicador de fuertes diferencias en cuanto a formación. El coeficiente de Gini del primer curso de programación podría servir como indicador del nivel de heterogeneidad para el siguiente curso.

La Figura 7 muestra la evolución temporal de los coeficientes de Gini para las poblaciones Est y Apr . Sobre Est los valores de los coeficientes se sitúan alrededor de 0,5, indicando poca variabilidad en el nivel de desigualdad y un tipo de desigualdad media. La situación es muy distinta sobre la población de aprobados. En este caso el coeficiente de Gini se sitúa alrededor de 0,1, indicando un nivel bajo de desigualdad.

Análisis. Con relación al total de estudiantes (Figura 6a) vemos que casi siempre el 10% alto tiene una nota próxima a 9. El 40% central está aproximadamente comprendido entre 4,5 y 7. Para el 50% aunque a veces incluye notas más altas que 3 la media es inferior a 1,5. Como ya hemos comentado la nota media de *Eb* está influenciada por la asignación de nota 0 a los no presentados. Apreciamos dos hechos importantes. Primero, la nota media del 50% inferiores muy baja (por debajo del 1,5), esto muestra un fenómeno de “cola larga” (long tail) que parece apartarse del comportamiento gaussiano. En una primera lectura la cola larga se puede interpretar como una tasa de fracaso alta, aunque no hay que olvidar que se trata de un curso de fase selectiva. Segundo, notemos que la nota de corte entre el 40% y el 50% oscila en la mayor parte de los casos alrededor del 3,5. Este hecho permite dar una explicación “sociológica” a posteriori del mecanismo de reevaluación. Este mecanismo está pensado como un incentivo para los estudiantes que están a caballo entre *Em* y *Eb*. Esto permite dar una contrapartida numérica al hecho de que los estudiantes reevaluables son los que “potencialmente pueden aprobar”. Es decir, se pretende incentivar a la clase media-baja y la baja-alta.

Cuando se trata sólo con aprobados *Apr* (Figura 6b), el 50% bajo no deja de estar alrededor del 6, aunque el 10% alto tiene una nota alrededor del 10 y el 40% central alrededor del 7. Las notas de corte en este caso parecen oscilar alrededor de las categorías tradicionales. Esto parece querer decir que la distribución (10%, 40%, 50%) y la distribución mediante notas convencionales capturan —aunque no de modo idéntico— el mismo tipo de fenómeno.

Analizamos ahora el coeficiente de Gini para nuestros datos. En el caso de todos los matriculados, el coeficiente de Gini es aproximadamente 0,5. Con carácter puramente orientativo comparemos con algunos datos en [14, Cap. 7], donde para la renta el coeficiente de Gini es de 0,19, y para la propiedad del capital es de 0,58. En economía un coeficiente de 0,5 se asocia a una desigualdad alta. Como primera aproximación podemos suponer lo mismo aquí. Sin embargo no hay que olvidar que este fenómeno está influenciado por la cola larga del 50% que hemos visto antes. Notemos que cuando sólo se tiene en cuenta el conjunto de los aprobados bajamos drásticamente a un máximo de 0,14.

Proponemos un análisis variacional en este contexto comparando la variación de la nota media con la variación del coeficiente de Gini. En este contexto un incremento de la nota media junto con un decremento del coeficiente de Gini indica que se ha conseguido incrementar la nota de forma más igualitaria (las notas estarán más cercanas a la media) mientras que si el coeficiente de Gini aumenta se detecta una mayor desigualdad en el reparto de las calificaciones.

	<i>Est</i>	<i>Ea</i>	<i>Em</i>	<i>Eb</i>	<i>Apr</i>	<i>Aa</i>	<i>Am</i>	<i>Ab</i>
t_4	0,54	0,06	0,18	0,66	0,11	0,02	0,06	0,04
t_5	0,47	0,03	0,13	0,57	0,12	0,01	0,05	0,05
t_6	0,43	0,02	0,11	0,55	0,11	0,01	0,04	0,06
t_7	0,49	0,04	0,14	0,62	0,14	0,01	0,05	0,04
t_8	0,46	0,04	0,11	0,63	0,11	0,02	0,04	0,05

Figura 7: Coeficientes de Gini.

Bajo este criterio se puede observar que, tanto en *Est* como en *Apr*, t_7 es la única medida que hace bajar la media y con la que aumenta la medida de la desigualdad en las dos clases. Reafirmando con este análisis la percepción, ya apuntada por el análisis coste-beneficio, de que las medidas adoptadas en t_7 han sido negativas. La medida t_8 contrarresta hasta cierto punto la medida previa sin llegar a alcanzar los niveles de t_6 .

5. Conclusiones

Este artículo presenta un análisis de la evolución temporal del curso Programación-1 a lo largo de diferentes semestres utilizando herramientas del área de la economía y de las ciencias sociales. El análisis coste-beneficio de la productividad se mide en función del trabajo de los docentes y de los resultados de los estudiantes, desagregándolos según diferentes categorías de notas. La distribución de los resultados también puede interpretarse desagregando los resultados de los estudiantes en el 10% alto, el 40% medio y el 50% bajo. Igualmente, las variaciones de estas distribuciones pueden caracterizarse usando el coeficiente de Gini.

El análisis del coste-beneficio desagregado por categorías de notas expande el realizado en [7] mostrando que el efecto de la productividad se extiende a todas las categorías y no se reparte uniformemente. Además, la ganancia marginal desagregada permite medir con más detalle la bondad de las diferentes medidas introducidas en el curso sobre de las diferentes poblaciones. Todo ello refuerza la tesis de que el estudio coste-beneficio de la productividad aumenta la comprensión de la evolución de los cursos y, por lo tanto, proporciona elementos para diseñar futuras estrategias.

El análisis de la desigualdad en los resultados muestra que a lo largo de los años el esfuerzo destinado a la mitad de nota más baja ha sido infructuoso. Si bien también se puede llegar a esta conclusión contando los suspensos, la clasificación (10%, 40%, 50%) lo muestra con claridad meridiana. La información que nos da el coeficiente de Gini se puede interpretar de varios modos pero sus valores cercanos a 0,5 parecen acordes a las expectativas de un curso de fase inicial.

Uno de los aspectos a tener en cuenta en el futuro es la incorporación de un tratamiento diferenciado de los estudiantes con *NP*. Este tratamiento diferenciado

requerirá por una parte un análisis detallado de los plazos y períodos en los que se produce el abandono. Esto permitiría una estimación más ajustada de la carga de trabajo por estudiante y de la carga total. La tarea no es fácil, se podría realizar una aproximación teniendo en cuenta además de las notas finales las calificaciones de todos los actos evaluatorios. Por otra parte se necesitaría un criterio adicional para determinar la distribución de la nota final entre este grupo que permita incluirlos dentro de la población de forma adecuada.

Referencias

- [1] K.J. Arrow. The economic implications of learning by doing. *The Review of Economic Studies*, 29(3):155–173, 1962.
- [2] M. Bauer, G.K. Smith, and J. Roberge. *Engaged Learning for Programming in C++: A Laboratory Course*. Jones and Bartlett Publishers, 2001.
- [3] L. Bot, P-B. Gossiaux, C-P. Rauch, and S. Tabiou. Learning by doing: a teaching method for active learning in scientific graduate education. *European Journal of Engineering Education*, 30(1):105–119, 2005.
- [4] S. Bowles. Towards an educational production function. In *Education, Income, and Human Capital*, pages 9–70. National Bureau of Economic Research, 1970.
- [5] L. Ceriani and P. Verme. The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *The Journal of Economic Inequality*, 10(3):421–443, 2012.
- [6] D. Diderot. Arithmétique politique. In *L'Encyclopédie*, 1751–1765.
- [7] M. Blesa, A. Duch, J. Gabarró, J. Petit, and M. Serna. Continuous Assessment in the Evolution of a CS1 Course: the Pass Rate/Workload Ratio. In *Computer Supported Education*, volume 583 of *Communications in Computer and Information Science*, pages 313–332. Springer, 2016.
- [8] O. Giménez, J. Petit, and S. Roura. Programació 1: A pure problem-oriented approach for a CS1 course. In *Informatics Education Europe IV*, pages 185–192, 2009.
- [9] C. Gini. *Variabilità e Mutabilità. Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche*. C. Cuppini, 1912.
- [10] E.A. Hanushek. Education production functions. In *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, 2008.
- [11] P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppälä. Review of recent systems for automatic assessment of programming assignments. In *10th Koli Calling Intl. Conference on Computing Education Research*, pages 86–93. ACM, 2010.
- [12] F.J. Martín-Carrasco, A. Granados, D. Santillan, and L. Mediero. Continuous assessment in civil engineering education: yes, but with some conditions. In *6th Intl. Conference on Computer Supported Education*, volume 2, pages 103–109. SciTePress, 2014.
- [13] J. Petit, O. Giménez, and S. Roura. Judge.org: an educational programming judge. In *43rd ACM Technical Symposium on Computer Science Education*, pages 445–450, 2012.
- [14] T. Piketty. *El capital en el siglo XXI*. RBA, 2015.
- [15] M.A. Revilla, S. Manzoor, and R. Liu. Competitive learning in informatics: The UVa online judge experience. *Olympiads in Informatics*, 2:131–148, 2008.
- [16] R.M. Solow. *Learning from 'Learning by Doing' Lessons for Economic Growth*. Stanford University Press, 1997.
- [17] J.E. Stiglitz and B.C. Greenwald. *Creating a Learning Society: A New Approach to Growth, Development, and Social Progress*. Columbia University Press, 2014.
- [18] V. Thomas, Y. Wang, and X. Fan. Measuring education inequality: Gini coefficients of education. Policy Research Working Paper no. 2525, World Bank, 2001.
- [19] V. Thomas, Y. Wang, and X. Fan. Measuring Education Inequality: Gini Coefficients of Education for 140 Countries (1960–2000). *Journal of Education Planning and Administration*, 17(1):5–33, 2003.
- [20] N.A. Tonin, F.A. Zanin, and J.L. Bez. Enhancing traditional algorithms classes using URI online judge. In *Intl. Conference on e-Learning and e-Technologies in Education*, pages 110–113, 2012.
- [21] H.R. Varian. *Intermediate Microeconomics: A Modern Approach*. W.W. Norton & Co., 2005.
- [22] E. Verdú, L.M. Regueras, M.J. Verdú, J.P. Leal, J.P. de Castro, and R. Queirós. A distributed system for learning programming on-line. *Computers & Education*, 58(1):1–10, 2012.