

## INFORMÁTICA

### Bandas de calificación de la asignatura

#### Nivel superior

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 14	15 - 28	29 - 39	40 - 51	52 - 63	64 - 75	76 - 100

#### Nivel medio

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 15	16 - 31	32 - 41	42 - 52	53 - 63	64 - 74	75 - 100

### Dossier del programa

#### Bandas de calificación del componente

#### Nivel superior

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 5	6 - 10	11 - 17	18 - 25	26 - 33	34 - 41	42 - 50

#### Nivel medio

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 6	7 - 13	14 - 20	21 - 27	28 - 35	36 - 42	43 - 50

### Ámbito que cubre el trabajo entregado y medida en que fue apropiado

La selección de problemas y la presentación de dossiers en esta sesión fueron buenos o muy buenos. Aún así hubo algunas elecciones inadecuadas, siendo la mayoría de éstas juegos o relacionadas con los juegos y de tipo “cerrado”. Los alumnos que eligieron un problema con final abierto obtuvieron, por lo general, una mayor puntuación. Los temas indefinidos son aquellos que se pueden simplificar fácilmente si demuestran ser demasiado estimulantes o ampliables si el tiempo lo permite.

Una vez dicho esto, una estrategia aún mejor es asegurarse desde el inicio de que el problema tiene un ámbito adecuado. Los problemas, obviamente, deberían poder resolverse

por una persona en un periodo razonable de tiempo. Los objetivos se deberían indicar clara y específicamente y, en la mayoría de los casos, cuatro objetivos llevan a un problema trivial, y ocho o más a uno de complejidad excesiva. Los objetivos deberían indicarse en relación a los requisitos de un usuario real, del cual se han obtenido datos relevantes de alguna forma. Los objetivos deben ser alcanzables y los alumnos ser capaces de explicar claramente al profesor cómo demostrar su éxito.

Las hojas de presentación también pueden proporcionar a los alumnos una lista de verificación útil para que se aseguren de que están siguiendo la estructura adecuada, incluida una tabla de contenidos. Estas hojas (5/PDCS) se pueden tratar al inicio del trabajo en el dossier.

Normalmente, los alumnos del nivel medio consiguen diez factores de dominio, aunque algunos aún añaden contenido sin sentido en sus programas sólo para intentar cumplir un elemento de dominio. Por ejemplo, un programa que sólo tiene un método adicional que imprime cadenas de texto en la consola no es suficiente para demostrar que se “domina” los métodos. Los alumnos deberían estar familiarizados con los métodos y, por tanto, en un dossier debería haber bastantes métodos.

Varios alumnos usaron matrices para almacenar datos, pero nunca almacenaron esos datos en un fichero. Es cuestionable si de esta forma se demuestra realmente el dominio de las matrices, ya que usarlas de esta forma conduce a una solución no usable.

Algunos alumnos del nivel medio no reciben puntos por el dominio de los indicadores/centinelas o de bibliotecas adicionales cuando parece que las podrían usar. Algunas bibliotecas adicionales son AWT o la interfaz Awt; algunas utilidades son StringTokenizer, ArrayList o LinkedList. El dominio de bibliotecas adicionales no se puede aplicar, sin embargo, al uso de java.io para la manipulación de ficheros. De la misma forma, el uso de ArrayList y de métodos asociados de ordenación no demuestra el dominio ni de las matrices ni de la ordenación. Para demostrar el dominio de este aspecto se deben usar matrices estáticas.

A menudo, los alumnos ofrecen pocos ejemplos, para confirmar la evaluación del profesor en C1, sobre la ejecución de sus programas y de que realmente funcionan como se indica. Esto es esencial para que el moderador confirme que se ha conseguido el logro de los aspectos.

Los factores de dominio deberían tenerse en cuenta al principio del proceso y, al menos 10 (preferiblemente 12) deben ser fácilmente identificables antes de que el profesor apruebe la propuesta del alumno. Los alumnos deberían ser guiados cuidadosamente en sus elecciones; no se les debería permitir que hicieran simplemente aquello que les resulta interesante. Se debería supervisar a los alumnos en cada fase del dossier, para que el profesor sepa cómo progresa cada uno y sean capaces de corroborar la afirmación de 5/PDCS que indica que sólo se considera trabajo aquél que realiza el alumno individualmente.

Los profesores deben asegurarse de que los alumnos entienden qué significa cada factor de dominio y deberían pedir asesoramiento, por ejemplo, en el CPEL. Sigue habiendo muchos problemas con los factores de dominio en el nivel superior, y los principales son:

El **Polimorfismo** se puede conseguir técnicamente sobrecargando métodos de interfaces de bibliotecas Java como `Object.toString()` y `ActionListener.actionPerformed(ActionEvent)`. Esto se considera trivial.

Ofrecer un constructor vacío y un segundo constructor para inicializar miembros dato de una clase también es trivial en la mayoría de los casos. El alumno debería justificar el uso de los constructores múltiples de alguna forma para demostrar que la solución se beneficia de ese diseño.

El **análisis sintáctico de un fichero de texto** u otro flujo de datos es algo más que usar un método o dos de las clases envoltorio (por ejemplo `Integer.parseInt(String)` y `Double.parseDouble(String)`). Esto es sólo un proceso de conversión de un tipo de datos a otro. El concepto implica tomar una cadena de texto, dividirla o descomponerla en caracteres dentro del flujo y producir datos para su posterior manipulación.

En los informes generales de la asignatura de mayo de 2006 (disponibles en el CPEL) se pueden encontrar más detalles sobre los aspectos individuales de dominio.

Los colegios deberían asegurarse de que los números de páginas sean los correctos cuando se indiquen las secciones en que se ha obtenido el dominio. Los moderadores no tienen la obligación de buscar solicitudes de dominio no documentadas; es responsabilidad del colegio ofrecer la información adecuada.

## Rendimiento alcanzado por los alumnos en cada uno de los criterios

### ANÁLISIS

La sección de Análisis es la base sobre la que se asienta un buen diseño y se crea una solución eficaz. Los alumnos que realizan la programación en primer lugar y luego hacen ingeniería inversa en el análisis suelen obtener calificaciones bajas en esta sección y en el resto del dossier.

#### Criterio A1

Los alumnos que carecen de un usuario y un problema reales se sitúan inmediatamente en desventaja en casi todo el dossier. Muchos alumnos siguen empezando por hablar del programa que van a desarrollar en lugar del problema de información que están investigando.

No es muy creíble afirmar que un gran banco u hospital están usando registros en papel y que han solicitado a un adolescente que cree una aplicación para ellos. Hay muchos problemas de pequeñas dimensiones adecuados en la mayoría de colegios internacionales en los que el usuario individual es accesible y en los que se puede ver la información existente.

Los problemas de juegos siguen siendo difíciles para la mayoría de alumnos, que suelen dedicar poco tiempo a la programación y bastante a las interfaces gráficas de usuario, las imágenes y el juego en sí. Cuando un estudiante realice un juego con imágenes, como cartas y piezas de ajedrez, se deberían conocer las fuentes.

Los mejores ejemplos definen el escenario y ofrecen alguna información para que el moderador pueda asimilar rápidamente el esbozo del tema principal.

Debería haber pruebas de la obtención de datos. Esto significa que se deberían mostrar datos reales: números y textos reales del dominio del problema. Muchos dossiers contienen

explicaciones, por ejemplo, "nombre y número de teléfono", pero debería mostrarse también datos de prueba, como por ejemplo "Fred Thomas, 312-4567".

El caso ideal es aquél en que se está mejorando un pequeño sistema manual y los documentos existentes en dicho sistema no tienen ningún valor al ofrecer datos de prueba. Los estudiantes de informática deberían ser expertos en el uso de escáneres y cámaras digitales para capturar e incluir datos relevantes: la mayoría del resto de estudiantes ya lo son.

### **Criterio A2**

Los alumnos deberían hacer referencia a los objetivos del análisis. Esto requiere algún tipo de explicación de por qué cada objetivo es importante. Por ejemplo:

*“Permitir que el usuario busque una fecha de nacimiento concreta, de forma que el jefe pueda saber quién cumple años ese día y enviarle una tarjeta de felicitación al empleado”.*

Como se indicó anteriormente unos seis objetivos de este tipo pueden crear un proyecto de un ámbito razonable.

Algunos estudiantes no discutieron las limitaciones en el uso de memoria, la respuesta temporal, la manipulación de errores y los requisitos del sistema. A menudo, no se discutió en absoluto sobre la usabilidad; el término "facilidad de uso" suele requerir una explicación en el contexto del problema.

En esta sección es preferible contestar por puntos en lugar de hacerlo como si fuera un ensayo, de lo contrario resulta difícil de leer y de referenciar en secciones posteriores.

### **Criterio A3**

A menudo, no incluyeron un diseño inicial del prototipo o no lo comprendieron correctamente. El diseño inicial puede ser un diagrama de flujo de datos o un esbozo muy básicos. El prototipo debería ser algo adecuado para discutir con el usuario final; preferentemente una interfaz de usuario, pero también debería incluir datos de muestra, no sólo elementos de menú. La mayoría de estudiantes ubican un diseño inicial en la sección A1, aunque es preferible que anteceda al prototipo en A3.

Algunos estudiantes presentaron un prototipo pero no un diseño inicial, que podría haber sido un simple esbozo o un simple diagrama de flujo de datos. El prototipo debería ser algo adecuado para discutir con el usuario final; preferentemente una interfaz de usuario, pero también debería incluir datos de muestra, no sólo elementos de menú.

La discusión con el usuario final debería estar documentada de alguna forma, y esto no siempre se hizo o resultó más bien trivial, como "Al usuario le gustó lo que vio" y afirmaciones similares. No parece muy probable que el usuario no tenga ninguna pregunta cuando el programador vuelve a él con una propuesta de solución a su problema.

## **DISEÑO**

Parece que la sección de diseño sigue causando la mayoría de los problemas a los alumnos, y esto suele deberse a que no se ha realizado un análisis exhaustivo del problema.

**Criterio B1**

Se deberían discutir e ilustrar estructuras de datos con bocetos y datos de muestra del dominio. Los alumnos peor preparados obtuvieron pobres resultados en esta sección, ya que no trataron ninguna de estas estructuras. Los diagramas abstractos de listas y árboles no son útiles en esta sección.

Si el alumno decide implementar características de la POO, es en esta sección en la que debería aparecer alguna discusión sobre la necesidad del polimorfismo o la herencia.

Algunos alumnos del nivel medio describieron las matrices para almacenar datos, pero omitieron el uso de un fichero para almacenar los datos permanentemente. Esto no representa una solución adecuada para un problema de bases de datos orientada a objetos, que suele ser el tipo de problema más común. La ausencia de almacenamiento permanente conlleva a una solución en absoluto realista e inutilizable, así como a dificultar la prueba en D1. Los dossiers sin fichero de datos tienden a presentar "listas" de datos que contienen sólo 1 o 2 registros. Tal como se discutirá posteriormente, esto crea problemas a los moderadores.

Los estudiantes deberían evitar extensas explicaciones sobre sus estructuras de datos y deberían usar diagramas para aclarar y transmitir eficazmente su pensamiento. Ésta suele ser una buena oportunidad para una revisión por los compañeros.

**Criterio B2**

El problema más común suele ser cortar, pegar y tal vez realizar alguna búsqueda / sustitución en el listado de código original. De esta forma no se obtienen puntos y es una pérdida del valioso tiempo del alumno.

Cuando los estudiantes deciden que el esfuerzo invertido en generar una solución completa con pre y post condiciones, parámetros y valores devueltos no merece los 2 puntos asignados, deberían describir la mayoría de los algoritmos principales; tal vez usando una descripción en pseudocódigo con comentarios preliminares. Ésta es una estrategia razonable para los estudiantes menos preparados.

No tiene mucho sentido intentar diseñar aquí algoritmos relacionados con la implementación de interfaces gráficas de usuario. Una mejor aproximación es usar instrucciones de entrada/salida donde sea pertinente y dejar los detalles hasta la fase de implementación. Parecidos comentarios se pueden aplicar a la manipulación de excepciones, caso de implementarse.

La planificación y descripción previas de algoritmos facilita escribir un programa funcional sin perder el tiempo.

**Criterio B3**

Ésta también debería ser una sección de diseño, y no una descripción de las clases usadas en la solución. Teniendo en cuenta que los alumnos podrían desviarse del diseño original, bastaría con un conjunto relativamente simple de módulos que dejaran claras las conexiones entre los algoritmos y las estructuras de datos. En caso de duda, un diagrama es una aproximación más simple y mejor para la mayoría de los alumnos que una aproximación en papel.

Algunos alumnos no incluyen enlaces a estructuras de datos y algoritmos.

## EL PROGRAMA

### Criterio C1

Los listados de programas suelen ser buenos y bastante centrados en obtener un buen estilo. Los alumnos deben asegurarse de que comentan el código fuente. Algunos alumnos del nivel medio escribieron todo el programa en un solo método. Esta opción no recibe ningún punto de dominio en el uso de algoritmos, y probablemente hace más difícil escribir el programa. Sería más beneficioso que descompusieran el programa en métodos.

Para facilitar la labor del moderador a la hora de confirmar los puntos otorgados por el profesor, las clases deberían iniciarse en una nueva página, o al menos estar claramente separadas de las clases precedentes, así como tener una descripción al inicio de las mismas. Una lista de clases con sus respectivos números de página al inicio del código resultaría enormemente útil.

Esto también servirá para las frecuentes clases en que el alumno ha indicado los números de página de los aspectos de dominio que son incorrectas y el profesor no las ha revisado aparentemente.

### Criterios C2 y C3

Las secciones de usabilidad y de la manipulación de errores han mejorado bastante en esta sesión, y la mayoría de alumnos ofrecen descripciones completas con ejemplos o crean una tabla de referencias a capturas de pantalla o listados de código, según sea necesario.

En la sección de usabilidad los alumnos que hacen referencia explícita a los criterios de usabilidad de la sección A2 obtuvieron más puntos que aquellos que no lo hicieron. Esto también demuestra la necesidad de tener criterios específicos en lugar de generales para la usabilidad. Por ejemplo la afirmación *“mi programa será fácil de usar para el usuario”* es muy difícil de probar, mientras que *“los botones de navegación estarán ubicados en el mismo sitio en todas las pantallas”* es más fácil de conseguir y demostrar.

### Criterio C4

Un problema clave de esta sección reclamar que un programa funciona bien y como se espera cuando se han presentado pocas capturas de pantalla o poco convincentes (a veces ninguna) en la sección D4.

Algunas capturas de pantalla poco convincentes son, por ejemplo, aquellas con las que se intenta convencer de que se ha implementado una búsqueda/clasificación o que se puede añadir/borrar registros en ficheros/listas/matrices, pero que realmente sólo muestran el mismo registro en distintas capturas.

Lo normal es que el profesor ejecute el programa con el alumno y confirme que la salida presentada en la sección D1 se puede obtener tal y como se reclama. Por tanto, si no hay apenas salidas en D1 el profesor debería animar a los alumnos a presentar más antes de otorgar la máxima puntuación aquí.

Una aproximación sólida que hace pensar al estudiante en realizar una sección D1 eficaz es proporcionar una descripción de cómo el programa cumple cada objetivo, incluyendo la salida impresa que demuestre que el programa funciona realmente como se espera.

## DOCUMENTACIÓN

Los estudiantes que hayan completado una solución total o parcialmente práctica no deberían descuidar esta sección, ya que muchas reclamaciones de consecución de logros en cuestión de criterios y aspectos de dominio requieren pruebas en esta sección.

Se recomienda encarecidamente que cuando un alumno no sepa compilar código para generar capturas de pantalla, el profesor le ayude en la compilación y en la generación de dichas capturas. El profesor, por supuesto, puede pedir al estudiante que clarifique qué código ha sido modificado por el profesor; estas partes del código no deben recibir ningún punto.

### Criterio D1

Como se recomendó anteriormente, se debería prestar más atención a demostrar que la solución funciona con ejecuciones normales de prueba y datos normales. Ofrecer diez páginas de cuadros de diálogo que muestren, por ejemplo, la introducción incorrecta de una contraseña o un fallo en la comprobación de un rango, seguido de una página en que se añade un registro a una matriz, dificulta enormemente la concesión de los criterios apropiados para C1 y D1 y los aspectos de dominio solicitados, muchos de los cuales requieren pruebas.

Algunos alumnos realizan cientos de páginas en esta sección. Deberían abstenerse. Sería mucho mejor tomar cuidadosamente capturas de pantalla que muestren que se ha conseguido cada criterio de A2 y que todos los aspectos de dominio que se reclaman están correctos.

La prueba de salida no válida no se debería pasar por alto, aunque quizá debería constituir el 33% de las capturas de pantalla en lugar del 90%. Es mejor concentrarse en lo que el alumno ha conseguido.

### Criterio D2

Las evaluaciones están mejorando globalmente, y a menudo siguen una aproximación estructurada, tratando los criterios de A2 en la sección sobre eficacia.

Muchos estudiantes, sin embargo, se limitan a responder a las preguntas que se muestran en la guía, pero realmente no muestran evidencias de que hayan pensado en profundidad sobre el rendimiento de su programa. Muchos alumnos indicaron la funcionalidad Web como un posible diseño alternativo. En algunos casos era realista, pero en otros no tiene sentido. Las sugerencias para mejoras y un diseño alternativo deberían ser sensatas.

### Criterio D3

Hay pocos problemas con el logro de esta sección, aunque algunos alumnos encuentran dificultades. Una aproximación útil es copiar capturas de pantallas y ofrecer un breve esbozo de los datos que se deben introducir. Para conseguir la máxima nota es necesario ofrecer instrucciones sobre la instalación.

Algunos alumnos de varios colegios escribieron extensas instrucciones acerca de cómo instalar e iniciar el programa, pero ninguna sobre cómo usarlo. Esto nos lleva a preguntarnos qué significa "ejecutar el programa", aunque en el futuro los estudiantes deberían recibir

instrucciones sobre cómo usar el programa, ya que esto es lo que el usuario necesita realmente.

## Sugerencias y recomendaciones para la enseñanza de alumnos futuros

Consulte la guía de la asignatura en lo que respecta al formato y otros requisitos del dossier del programa. Algunos dossiers no siguieron el formato obligatorio, no usaron el lenguaje Java o no incluyeron números de páginas y tablas de contenidos. Todo esto dificulta la moderación y expone a los estudiantes a un alto riesgo de perder puntos. Un problema adecuado para el dossier de programa podría:

- Incluir un usuario que sea una persona real y accesible.
- Tratar algún tema conocido por el alumno.
- Haber sido cuidadosamente evaluado por el profesor.
- Tener el potencial de satisfacer suficientes aspectos de dominio del nivel superior.
- Haberse investigado y analizado.

Una estrategia para conseguir estos objetivos es hacer que el alumno exponga sus problemas ante sus compañeros para que los comenten.

Esto puede hacerles pensar antes de empezar a trabajar y a su vez da la oportunidad al profesor de evaluar la adecuación de la tarea justo al principio, cuando se toman muchas decisiones cruciales. En este momento, los alumnos deberían recibir una copia del formulario 5/PDCS y una lista de comprobación para que sepan exactamente lo que se requiere que hagan, y poder trabajar de esta forma en la estructura del informe a medida que avanzan. Algunos dossiers dan la impresión de haber sido hechos con prisa, como si el alumno se hubiera dado cuenta de repente de la necesidad de incluir números de páginas, tablas de contenidos y similares en la parte final del proceso. La esencia del dossier del programa es una planificación adecuada y el seguimiento del ciclo de las fases de análisis, diseño, implementación y documentación. Los profesores deberían asegurarse de realizar todo el trabajo administrativo para evitar el retraso en la concesión de notas a sus alumnos.

## Nivel superior Prueba 1

### Bandas de calificación del componente

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 15	16 - 30	31 - 41	42 - 51	52 - 60	61 - 70	71 - 100



## Áreas del programa o del examen en que los estudiantes demostraron estar bien preparados

Un problema en curso es el hecho de que los estudiantes no leen las preguntas detenidamente y ofrecen respuestas generales que con frecuencia no responden a la pregunta.

La pregunta 17 requería que los alumnos apreciaran la cuestión fundamental del procesamiento de datos en una lista, y aunque respondieron bien al núcleo de la pregunta, muchos no parecieron entender la diferencia entre una estructura de datos dinámica y una estructura estática.

En la pregunta 19 (a) no comprendieron bien el papel del contador de programa. El contador de programa almacena la dirección de la siguiente instrucción que se va a ejecutar. Además, dieron respuestas pobres a la pregunta sobre el papel de una pila para evaluar expresiones matemáticas.

La pregunta 21 demuestra que bastantes alumnos tuvieron dificultades a la hora de construir algoritmos. El desarrollo de algoritmos es fundamental en informática y seguirá siendo objeto de evaluación.

La pregunta 22 planteó muchos problemas a los estudiantes. Al ser la última pregunta, puede que los estudiantes respondieran con prisa. Los conocimientos sobre cilindros fueron limitados y los alumnos no comprendieron bien la alusión al acceso paralelo a los datos, en lugar del acceso secuencial.

## Niveles de conocimiento, comprensión y habilidad mostrados

Globalmente, los estudiantes mostraron unos buenos conocimientos generales. Resultó gratificante ver muestras de buenos conocimientos sobre el contenido del curso, como: lógica booleana, recursividad y representación de números.

## Puntos fuertes y débiles de los estudiantes al abordar las distintas preguntas

### 1 Informe de viabilidad

En general respondieron bien, aunque sólo unos pocos alumnos se centraron en los detalles de bajo nivel de los programas.

### 2 Mantenimiento de sistemas

Respondieron bien a esta pregunta.

### 3 Datos de prueba

Algunos alumnos se centran sólo en datos no válidos en lugar de hacerlo en los datos normales, no válidos y extremos.

### 4 Intérprete vs compilador

Parece que un número de alumnos no es consciente del proceso fundamental de traducir desde código de alto nivel a un código de bajo nivel que pueda ejecutarse.

## 5 Acceso directo

La mayoría de los alumnos indicaron claramente que se podía acceder sin referenciar a otros miembros del fichero o de la lista. Otros, sin embargo, confundieron los conceptos de registro de un fichero y elemento de datos de una lista.

## 6 Aplicación de pantalla táctil

Respondieron correctamente a esta pregunta.

## 7 Conversiones binaria y hexadecimal

Los alumnos demostraron capacidad para manipular expresiones binarias y hexadecimales.

## 8 Defecto

Esta pregunta planteó algunas dificultades para varios estudiantes. Se produce cuando un número es demasiado pequeño para ser representado; normalmente se define con un valor cero.

## 9 Expresión booleana a partir del diagrama de un circuito

Esta pregunta demostró que la mayoría de alumnos sabía derivar una expresión booleana y simplificarla, aunque muchos no simplificaron de  $AB + B$  a  $B$ .

## 10 Funciones de los registros de la CPU

Esta pregunta tiene un nivel de dificultad considerable. El registro de instrucciones almacena la instrucción que se ejecuta actualmente. El acumulador almacena los resultados temporales de la aritmética de los cálculos lógicos en la unidad aritmética lógica.

## 11 Algoritmo recursivo

Respondieron bien a la pregunta 11 (a), siendo muchos los estudiantes que se dieron cuenta de que un algoritmo recursivo se llama a sí mismo y que debe tener una condición de parada. En la parte (b), la de la traza, también respondieron bien.

## 12 Procesamiento en tiempo real en una estación de energía nuclear

Respondieron bien a esta pregunta.

## 13 Conversiones de decimal a hexadecimal y binario

La pregunta 13 demostró que algunos estudiantes no saben aplicar la representación en complemento a dos.

## 14 Software de defragmentación

Esta pregunta mostró que los estudiantes sabían que los bloques de datos de un fichero se pueden dispersar y que la defragmentación los une de forma contigua y, por tanto, reduce el tiempo de acceso. No obstante, varios estudiantes indicaron también que se obtendría un mayor espacio en disco, lo cual no es cierto.

## 15 Polimorfismo

En la pregunta 15 muestran un agradable aumento en el nivel de conocimientos. Las respuestas más comunes indican el uso de nombres de métodos que fueron seleccionados a través de la búsqueda de patrones de los parámetros, implementados a través de varios constructores de objetos.

### **16 Complejidad O Mayúscula para el algoritmo quicksort**

Respondieron bien a la pregunta 16.

### **17 Lista ordenada de nombres en memoria principal**

- (a) Muchos alumnos no hicieron referencia al papel de los apuntadores, pero fueron conscientes del hecho de que la inserción y la eliminación sería más fácil con apuntadores que usando una matriz.
- (b) Muchos alumnos usaron diagramas y demostraron un buen nivel de conocimientos sobre la función básica de la manipulación de apuntadores (referencias en Java) para insertar el nuevo nodo. En este nivel de detalle, los estudiantes deben saber que no deben sobrescribir apuntadores.
- (c) Muchos estudiantes indicaron el uso de un árbol binario; algunos, el de una matriz, lo que normalmente no es adecuado para un nivel de adición/eliminación debido a que la lista debe volver a ordenarse, situación que no se produce en un árbol.

### **18 Compañía de coches con un fichero de registros de stock**

- (a) La mayoría de estudiantes indicaron un acceso más rápido, pero pocos se dieron cuenta de que los registros no se tenían que mantener en orden.
- (b) En general respondieron bien.
- (c) Los estudiantes se centraron en las colisiones y a menudo no se preocuparon, por ejemplo de una mayor rapidez en los cálculos o del tamaño del espacio de hash.
- (d) (ii) La mayoría de estudiantes indicaron una colisión, pero hubo bastantes que no indicaron cómo se podría tratar mediante desbordamiento, encadenamiento o pruebas. Muchos estudiantes pensaron que se produciría un error de algún tipo y que el registro no se podría almacenar totalmente.

### **19 Pilas y llamadas a subprogramas**

- (a) El contador de programa almacena la dirección de la siguiente instrucción. Muchos alumnos mostraron que una instrucción push situaba la dirección devuelta en la pila y que una operación pop restauraba el estado, pero no indicaron referencias adecuadas al hecho de que la instrucción pop modifica el contador de programa.
- (b) Muchos estudiantes no indicaron que se usaba una pila y, por tanto, que la notación postfixa permite que la evaluación se realice de una sola pasada.
- (c) Respondieron bien.
- (d) Algunos estudiantes tuvieron problemas con esta pregunta y algunos diagramas no se parecían en absoluto a una pila. El error más común fue ubicar los elementos de la pila al revés (es decir, en la parte superior).

### **20 Red de área local de una pequeña compañía**

- (a) Los estudiantes mostraron sólidos conocimientos sobre la comunicación inicial entre dos dispositivos antes de la transmisión, y ofrecieron ejemplos sencillos.
- (b) Respondieron bien.

- (c) Respondieron bien, pero muchos no hicieron referencia al hecho de que se usa un conjunto de reglas para dirigir el funcionamiento de un cortafuegos.
- (d) Los estudiantes mostraron un buen nivel de comprensión sobre cómo salvaguardar datos de un acceso no autorizado. Las sugerencias más comunes fueron que la combinación de usuario/contraseña o la encriptación limitarían las posibilidades de un acceso al sistema.

## 21 Puertas lógicas y construcción de algoritmos asociados

Esta pregunta dividió a los estudiantes en dos grupos: los que pudieron responderla y los que no.

- (a) Respondieron razonablemente bien, pero un gran número de ellos parecía no saber construir una tabla de verdad o reconocer el operador OR exclusivo (XOR).
- (b) Los estudiantes ofrecieron una considerable variedad de algoritmos que funcionaban. El error más común fue indicar que  $A=0$  y  $B=0$  devolvían verdadero. Una respuesta común fue:

*If (a == b) return true*

Lo que es un XOR, ya que permite que 0,0 y 1,1 devuelvan verdadero como condiciones. La condición correcta es si  $(a==1 \text{ and } b==1)$  devuelve falso, de lo contrario devuelve verdadero.

- (c) Los estudiantes ofrecen respuestas variadas, y resulta agradable ver el nivel de conocimientos demostrado. Algunos enumeraron todas las posibilidades en una larga instrucción if, lo cual es aceptable. La clave era comprobar que si se producía un resultado falso, no podría existir la igualdad, y el algoritmo terminaría (devolvería false).

Varios estudiantes usaron la comparación *if(logica1(a,b) == logica2(a, b) return true*. Esto, sin embargo, no finaliza el bucle, ya que la primera igualdad terminaría la comprobación.

La clave es devolver falso en la comparación: *if (not (logica1(a,b) == logica2(a,b) )*

- (d) Muchos alumnos supieron reducir la expresión y demostraron entender la lógica booleana y su aplicación a las reglas.

## 22 Funciones del sistema operativo y formato de un disco duro

- (a) Respondieron razonablemente bien.
- (b) Entendieron bien los conceptos de pista y sector, pero la noción de cilindro no se entendió bien, tal como se muestra en la parte (c).
- (c) Muchos estudiantes intentaron una explicación técnica que no resultó adecuada. Los cilindros permiten la lectura y la escritura en paralelo, lo que permite que se puedan manipular más datos en cada ciclo.
- (d) Una aproximación por bloques reduce el número de ciclos de acceso de lectura y escritura; pocos estudiantes supieron dar esta respuesta.

## Recomendaciones y orientaciones para la enseñanza de futuros estudiantes

Los profesores aún deben seguir educando a los alumnos en el arte de leer las preguntas. Además, el número de puntos concedidos a una pregunta indica el número de puntos que se esperan en la respuesta. Los algoritmos tienen que ser cruciales para la preparación del examen, y los alumnos deberían practicar simulando las condiciones de un examen.

## Nivel superior Prueba 2

Nota: Hubo un problema con la pregunta del estudio de caso, a la que se asignaron sólo 30 puntos, en lugar de los 40 obligatorios. Esto hizo que el examen se puntuara sobre 90 y que se ajustara el factor de escala.

### Bandas de calificación del componente

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 15	16 - 31	32 - 38	39 - 47	48 - 56	57 - 65	66 - 90

## Áreas del programa o del examen en que los estudiantes demostraron estar bien preparados

Hubo gran variedad en el nivel de trabajo. Aunque hubo excelentes scripts, demostraron buenos conocimientos sobre informática y familiaridad con la terminología, fue una de las áreas en que estaban peor preparados.

## Niveles de conocimiento, comprensión y habilidad mostrados

Los alumnos tuvieron dificultades para centrarse en las preguntas de discusión, escribiendo bastante más, sin centrarse en las partes importantes, de lo que sugería la asignación de puntos.

La mayoría de estudiantes tenían conocimientos suficientes para afrontar todas las preguntas. A menudo, resultaba evidente que los alumnos tenían conocimientos y entendían el tema, pero la discusión no era suficiente para obtener la máxima nota.

En todas las preguntas hubo algunas respuestas excelentes, demostrando que la cobertura del programa de estudios es magnífica en algunos colegios.

## Puntos fuertes y débiles de los estudiantes al abordar las distintas preguntas

### 1 Listas enlazadas y algoritmos recursivos

La mayoría de alumnos que respondieron a esta pregunta realizaron una buena traza del código. Algunos alumnos no etiquetaron claramente la cabecera y los enlaces entre los nodos. Algunos alumnos mostraron la lista en orden inverso. La mayoría de alumnos supo indicar la eficiencia de añadir nodos a la lista y de buscar en la lista.

La mayoría también hizo un intento razonable de trazar el método recursivo y algunos respondieron perfectamente.

Las respuestas sobre construir un método que devuelva verdadero si el valor está en la lista y falso en el caso contrario variaron, de mediocres a excelentes.

Por lo general, responden bien a cómo se podría realizar una búsqueda en un árbol binario.

## **2 Aritmética binaria y representación en coma fija y flotante**

Ésta es la pregunta a la que mejor respondieron. La mayoría de estudiantes supo representar números en formato binario y convertirlos de decimal a binario. Algunos no supieron calcular el valor del número representado por la mantisa y el exponente.

La mayoría esbozó con claridad un ejemplo de aplicación informática que usara valores en coma flotante, e indicó por qué los errores suponen un problema en la representación.

## **3 Métodos de organización de ficheros para una agenda telefónica**

Pocos alumnos respondieron realmente bien a esta pregunta, pero la mayoría ofreció buenas respuestas a alguna parte de la misma. Los alumnos que obtuvieron menos puntos fueron aquellos que ofrecieron una alternativa incorrecta en la parte (b). Algunas no eran ni siquiera secuenciales y muchas hicieron uso de un árbol binario o una lista enlazada. Un problema similar surgió en las partes (f) y (g).

Parece que los alumnos no saben mucho sobre los métodos de organización de ficheros. Muchos usaron terminología incorrecta como, por ejemplo, confundir fichero y registro. Los alumnos tienen muchas lagunas en cuanto a la organización de ficheros, aunque éste es un tema extenso en el programa de estudios del nivel superior.

## **4 Pregunta del estudio de caso: computadores y discapacidad**

Los alumnos, por lo general, contestaron bastante bien a esta pregunta, algo previsible teniendo en cuenta que tuvieron tiempo suficiente para preparar el examen

Las partes (a) y (b) fueron a las que mejor respondieron. Algunos tuvieron en cuenta cuestiones relacionadas con la discapacidad en dispositivos de salida en lugar de los dispositivos de entrada especificados en la pregunta.

La mayoría pudo esbozar una estructura de datos que almacenara abreviaturas y la frase completa y fue capaz de sugerir un método para acceder a la abreviatura.

Las partes (c), (d) y (e) estaban relacionadas con las estructuras de datos y, por tanto, muchos alumnos respondieron bien, consiguiendo casi la máxima puntuación.

La mayoría respondió bien a las partes (f) y (g).

En la parte (h), muchos alumnos se limitaron a reiterar el problema general, pero no ofrecieron las consecuencias sociales relacionadas con el aumento de la dependencia de la sociedad en las redes de computadores. Además, muchas respuestas se desviaron del punto principal, que era la **falta** de acceso a las redes.

## Recomendaciones y orientaciones para la enseñanza de futuros estudiantes

Los alumnos, especialmente los peor preparados, deberían dedicar más tiempo a trazar y construir algoritmos. Necesitan adquirir confianza para comprender y escribirlos en condiciones similares a las de un examen.

Cuantos más exámenes practiquen los alumnos, mejor. En el proceso de enseñanza/aprendizaje deberían incluirse preguntas de exámenes anteriores. Esto ayudará a reforzar el tema y a que se familiaricen con la estructura de las preguntas y con la forma de preguntar en un examen.

Se debería emplear más tiempo en usar el estudio de caso como herramienta de enseñanza en contextos diferentes y no sólo en el escenario proporcionado.

## Nivel medio Prueba 1

### Bandas de calificación del componente

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 12	13 – 24	25 - 29	30 - 36	37 - 43	44 - 50	51 - 70

### Áreas del programa o del examen en que los estudiantes demostraron estar bien preparados

Está claro que algunas escuelas no preparan bien a sus alumnos, si es que los preparan algo, para los exámenes teóricos. Muchos alumnos tienen pocos o ningún conocimiento para realizar el examen. En algunos casos los alumnos fueron incapaces de terminar ninguna pregunta, salvo las de programación y las relacionadas con números.

La nota más baja fue un cero y es verdaderamente alarmante pensar que en un colegio ni siquiera se recomiende un libro o un sitio Web sobre informática antes de someter a los alumnos a semejante suplicio.

### Niveles de conocimiento, comprensión y habilidad mostrados

Aunque algunos alumnos mostraron un nivel bastante alto de conocimientos y comprensión, resulta evidente que muchos están muy poco preparados para este examen.

Parece que los estudiantes tienen menos conocimientos sobre hardware que sobre temas de programación lógica y experiencia del usuario final.

De muchas respuestas se deduce que los alumnos podrían haber respondido a la pregunta si hubieran pensado más en lo que se estaba preguntando, en cómo obtener más puntos y en responder con mayor precisión. Es evidente que muchos alumnos no leyeron las preguntas y fueron incapaces de usar correctamente la terminología.

Algunos intentaron dar varias respuestas a las preguntas, y esto es una estrategia que se debería evitar. Parece que muchos estudiantes tuvieron problemas para comunicarse con el examinador. Asimismo, en algunos casos se deberían mejorar las habilidades de escritura practicando con exámenes anteriores.

## Puntos fuertes y débiles de los estudiantes al abordar las distintas preguntas

### 1 Informe de viabilidad

Los alumnos podían identificar muchos componentes, y bastantes de ellos pudieron identificar al menos dos. Un error común fue incluir elementos que se podrían haber generado en posteriores análisis del problema como por ejemplo técnicas de búsqueda o requisitos de hardware. Hubo menos respuestas referentes a sistemas que ya estaban en funcionamiento.

### 2 Intérprete vs compilador

La mayoría de alumnos supo responder correctamente a esta pregunta. Aunque Java y los IDE modernos pueden confundir este tema (compilando a bytecode que posteriormente se interpreta en tiempo de ejecución, mientras que el IDE indica los errores de ejecución), la mayoría de los alumnos aún reconoce las diferencias esenciales entre estos programas.

### 3 Registro

Muchísimos estudiantes confunden un registro con el proceso de “registrar” aplicaciones en un sistema operativo. Otros estudiantes supieron describir la función de los registros en relación con la pregunta siguiente sobre el ciclo de instrucciones máquina. Los mejores estudiantes supieron explicar la relación entre registros, palabras y buses en la UCP.

### 4 Ciclo de instrucciones

Algunos alumnos usan el término dato en lugar de instrucción: obtener datos, decodificar datos, etcétera. Otros confunden este ciclo con el modelo de entrada-procesamiento-salida de los sistemas informáticos. Algunos estudiantes indicaron los pasos en lugar de esbozarlos. Se debería prestar atención a los verbos de acción.

### 5 Router

Muchos alumnos identificaron un router con un dispositivo que conecta una red a Internet. Aunque puede darse el caso de que un dispositivo actual combine las funciones de un módem de banda ancha y un router inalámbrico, deberían conocer las diferencias entre ambos.

### 6 Conversiones binaria y hexadecimal

La mayoría respondió muy bien. Causó problemas principalmente a aquellos a los que nunca se les había explicado nada del tema. Hubo pocas respuestas “a medias”.

### 7 Datos analógicos y digitales

Muchos alumnos supieron distinguir correctamente entre datos binarios y analógicos y comprendieron la necesidad de la conversión o supieron dar ejemplos de conversiones



necesarias de datos analógicos. No obstante, no ofrecieron prácticamente ningún ejemplo ni mencionaron otro método de conversión.

### **8 Errores en la introducción de datos**

Esta pregunta no supuso ningún problema para la mayoría de alumnos.

### **9 Utilidad para la compresión de datos**

Esta pregunta no supuso ningún problema para la mayoría de alumnos.

### **10 Matrices**

Muchos alumnos demostraron facilidad con las matrices e índices y la mayoría respondió correctamente a esta pregunta.

### **11 Integridad de los datos**

El error más común fue dar respuestas en relación con la encriptación, y los alumnos no siempre indicaron conocer la diferencia entre la integridad y la seguridad de los datos. Algunos alumnos confundieron los dígitos de comprobación con las sumas de comprobación. Algunas respuestas hicieron referencia al envío de datos por duplicado para comparar archivos, lo cual puede deducirse a partir de una lectura informal de la pregunta, que hace referencia al momento “durante la transmisión” de datos.

### **12 Discos duros**

Aunque no se espera que los estudiantes del nivel medio conozcan detalles sobre el formateo de los discos duros en sectores, pistas y cilindros, cualquier discusión sobre memoria secundaria y tipo de acceso debería sacar a la luz algunas de las características, que permiten el acceso directo. Por ejemplo: el movimiento de la cabeza a cualquier parte del disco, la posibilidad de leer un bloque sin atravesar otras partes del medio y la inclusión de una tabla de direcciones dentro de los medios de acceso directo de cualquier tipo.

### **13 Traza de un algoritmo para insertar espacios entre cadenas de texto**

La mayoría de alumnos comprendió y respondió bien. Algunos hicieron suposiciones más que realizar una traza eficaz del código. Por ejemplo, asumieron que los asteriscos formaban parte de algún sistema de introducción de contraseñas. Bastantes alumnos también entendieron que se podría usar para alinear columnas de texto en una tabla.

### **14 Empresa con ficheros de stock en varias ciudades**

En la parte (a) muchos estudiantes pusieron ejemplos de verificación, en lugar de validación. La mayoría de ellos obtuvo puntos en las partes (b) y (c). La mayoría respondió bien a la pregunta sobre actualización de ficheros, mientras que sólo la mitad de los que lo intentaron respondió bien a la sección del final. Algunos sugirieron que la red no se utilizaba en absoluto; e.g. enviando faxes y telefoneando a las sucursales, aunque ésta no parece ser una respuesta adecuada al problema

### **15 Cirugía: oficina y redes**

De nuevo, la pregunta parece resultar clara para muchos alumnos. Aunque no respondieron muy bien a la parte (b), muchos no sabían cuál era el componente hardware o si lo sabían se limitaron a nombrarlo (pasarela/router/módem) sin explicar sus funciones. Casi todos los

alumnos supieron responder a las preguntas (c) y (d) sin dificultad, ya que el tema se había tratado en exámenes recientes.

El término cirugía resultó una fuente potencial de confusión para algunos alumnos y, por tanto, se rebajó la puntuación, aunque esto no tuvo mayores consecuencias. A la hora de discutir los beneficios y riesgos para los pacientes respondieron bien a la cuestión de las operaciones quirúrgicas.

### **16 Grabador de DVD y microprocesador**

Hay alguna evidencia de que algunos estudiantes no están familiarizados con el funcionamiento de un grabador de DVD, y algunos se equivocaron a partir de la respuesta (a) de la primera parte de la pregunta. Tal vez pensaron que era una pregunta con trampa (algo que los examinadores evitan).

La mayoría de alumnos supo identificar las entradas correctas aunque, de nuevo, el control manual apareció (incorrectamente) en muchas respuestas.

Muchos supieron reconocerla como una aplicación en tiempo real y supo identificar las características fundamentales del procesamiento en tiempo real, incluso aquellos alumnos que no respondieron bien a las preguntas (a) y (d) pudieron obtener en muchos casos los puntos correspondientes.

## **Recomendaciones y orientaciones para la enseñanza de futuros estudiantes**

Los profesores deberían hacer más énfasis en las secciones del currículum no relacionadas con la programación. Aunque la habilidad de programar es necesaria para completar correctamente el dossier, debemos reconocer que el hardware, la arquitectura y las redes también son áreas importantes que deben estudiarse.

La terminología no es una parte especialmente excitante del curso, pero los estudiantes tienen pocas posibilidades de éxito a menos que la conozcan bien. Por tanto, se deberían usar métodos interesantes y competitivos para animar a los estudiantes a que memoricen la jerga informática.

Los profesores también deberían animar a los alumnos a que realizaran exámenes de años anteriores antes de enfrentarse al examen real, así como reforzar técnicas de examen, como leer la pregunta, observar los verbos de acción y responder acorde con los puntos asignados.

Como se menciona anteriormente, se debería disuadir a los estudiantes de dar varias respuestas a una pregunta, pues sólo se marcará la primera respuesta, incluso aunque la respuesta correcta aparezca después.

Las respuestas sin cualificar sólo pueden ser aceptables cuando se usa el verbo de acción "indicar"; el resto debería cualificarse o compararse para obtener puntos, sobre todo cuando se está buscando una explicación o una descripción.

## Nivel medio Prueba 2

### Bandas de calificación del componente

<b>Nota final:</b>	1	2	3	4	5	6	7
<b>Puntuaciones:</b>	0 – 11	12 – 23	24 – 30	31 - 36	37 - 41	42 - 47	48 - 70

### Áreas del programa o del examen en que los estudiantes demostraron estar bien preparados

La estructura del examen se compone de tres preguntas, dos de las cuales contienen una gran cantidad de contenido algorítmico y una tercera que trata el estudio de caso. En este examen se obtuvo una amplia variedad de notas, siendo todas las preguntas accesibles para los mejores estudiantes. La mayoría de alumnos respondió bien la parte del estudio de caso.

Las notas más bajas no se debieron tanto a la dificultad de la pregunta del algoritmo como a la distinta forma de tratar los algoritmos y la resolución lógica de problemas en los diferentes centros, ya que había notables diferencias entre los resultados de los colegios. Aún hay centros que no preparan bien a sus estudiantes para lo que siempre es una parte importante de este examen. Al contrario, aquellos estudiantes de colegios que preparan bien a sus alumnos obtuvieron buenas puntuaciones, y los algoritmos fueron relativamente sencillos.

En relación al estudio de caso, las preguntas que empiezan con verbos de acción como “explicar”, “discutir” o “comparar” siguen siendo difíciles para muchos alumnos, que tienden a ofrecer sólo una respuesta descriptiva.

### Niveles de conocimiento, comprensión y habilidad mostrados

Tal como se indica anteriormente, las áreas que resultaron difíciles para algunos alumnos fueron las mismas en que otros (de colegios diferentes) mostraron un nivel considerable de conocimiento y comprensión. Los estudiantes de aquellos colegios que los preparan para pensar lógicamente en la construcción de algoritmos siempre obtendrán una elevada puntuación en este examen. Siempre se tratarán procesos comunes como la búsqueda, la ordenación, la inspección de valores de una matriz o la devolución de valores al método que llama. Muchos alumnos, por tanto, supieron construir algoritmos correctos desde el punto de vista lógico y, en muchos casos, elegantemente programados. La capacidad de responder rápidamente a estas preguntas también deja más tiempo a los estudiantes para la pregunta del estudio de caso.

### Puntos fuertes y débiles de los estudiantes al abordar las distintas preguntas

#### Pregunta 1

Las matrices son una estructura de datos que usan frecuentemente, y el acceso a sus contenidos mediante bucles (matrices unidimensionales) y bucles anidados (matrices bidimensionales) suele incluirse en las preguntas del nivel medio y del nivel superior. Los

estudiantes, sin embargo, aún cometen errores con los índices, en este caso muchos se equivocan en las etiquetas para otra columna de datos (en la parte (a)).

Los métodos requeridos en (b) y (c) fueron bastante claros, aunque resulta evidente que algunos colegios siguen ofreciendo una preparación insuficiente en cuanto a la anidación de bucles, ya que tienen problemas para finalizar las condiciones. Por ahora, queda claro que el índice inicial de una matriz es 0 y el índice superior la longitud de la matriz - 1.

De nuevo, la pregunta sobre la "media" en la sección (d) no era complicada, aunque algunos alumnos, lamentablemente, evitaron los bucles e incluyeron en su lugar líneas de sumas. Cuando se piden "mensajes de salida adecuados", se suele probar la capacidad de los estudiantes para incluir variables en el mensaje (en este caso, el piso correcto).

En 1 (e) se pregunta por qué no es adecuado usar una estructura de matriz. Esta pregunta resultó más difícil que escribir los algoritmos anteriores, y sólo los mejores alumnos ofrecieron respuestas correctas, la mayoría de las cuales trataban los métodos de acceso a los datos mediante índices y bucles.

## Pregunta 2

Aquí se introdujeron algunos elementos de la programación orientada a objetos, incluyendo conceptos que resultan difíciles para la mayoría de estudiantes y, por tanto, que se suelen tratar sólo en el nivel superior. No obstante, los alumnos del nivel medio deberían saber definir "constructores" (partes (a) y (b)). La pregunta pretendía diferencias entre los programadores más capacitados y el resto. Muchos no tienen del todo claro lo que supone crear un nuevo objeto de una clase concreta. Muchos estudiantes están familiarizados con el método principal, consistente en sólo una llamada al constructor, por ejemplo `Numeros()`; el mismo constructor que luego llama a los distintos métodos presentes en el programa (de 1 sola clase). Esto les lleva, por desgracia, a creer que la función de un constructor es ejecutar un programa (y no crear un nuevo objeto).

Las partes (b) y (c) fueron, de nuevo, algoritmos claros. Algunos cometieron en la pregunta (b) el error, al buscar el máximo, de no usar una variable temporal para almacenar el máximo, sino para comparar los valores sucesivos de la matriz. Se podría haber resuelto correctamente de esta forma, pero la mayoría se habría encontrado con un error en el rango de la matriz al solicitar un índice más que el superior de la matriz.

La mayoría identificó en (c) la búsqueda secuencial y ofreció una "búsqueda binaria" como mejor alternativa en (d).

Habiendo introducido los dos métodos en (b) y (c), la mayoría de alumnos hizo un intento razonable de incluirlas en el método principal de (e). Sólo aquellos estudiantes que conocían bien la programación orientada a objetos se dieron cuenta de que había que hacer referencia a los métodos de un objeto concreto (en este caso, el objeto "matriz") usando la notación punto.

## Pregunta 3

La mayoría de alumnos obtuvo un considerable número de puntos en la pregunta del estudio de caso. A los alumnos de los colegios en que se dedicó tiempo al estudio de caso les fue bastante mejor con menos respuestas "literales" tomadas del propio estudio de caso.

Las preguntas que usan los verbos de acción “discutir”, “sugerir” y “explicar” requieren algo más que una breve respuesta descriptiva. Los alumnos que respondieron de esta forma no obtuvieron la máxima puntuación para cada apartado.

En (a) e (i), por ejemplo, indicar que la persona tendría dificultades para usar un teclado y el ratón habría merecido 1 punto, pero si además se hubiera puesto un ejemplo de un diseño alternativo (mayores teclas, teclas agrupadas, etc) se habría conseguido otro punto. De igual forma, sugerir una aplicación del control biométrico en (b) requería tanto una descripción de la aplicación como una forma de conseguirla.

La parte (f) ponía fin al examen con una pregunta bastante difícil. Se preguntaba acerca de las “consecuencias del aumento en la **dependencia** que tiene la gente sobre los computadores”. Sólo los mejores alumnos respondieron en función de cómo estaba formulada, mientras que la mayoría trató el “aumento en el **uso** de los computadores”, que no es lo mismo.

## Recomendaciones y orientaciones para la enseñanza de futuros estudiantes

La resolución de problemas mediante algoritmos es una parte fundamental del plan de estudios y debería ser una parte importante de la enseñanza de esta materia. Las escuelas también deberían animar a sus alumnos a prestar especial atención al formato de cada pregunta, incluyendo aquellas que aparecen en el estudio de caso, para determinar exactamente qué se está preguntando y para que no se limiten a dar una respuesta descriptiva que responda sólo parcialmente la pregunta.