

INFORMÁTICA

Bandas de calificación de la asignatura

Nivel Superior

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 14	15 - 28	29 - 39	40 - 51	52 - 63	64 - 75	76 - 100

Nivel Medio

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 14	15 - 30	31 - 42	43 - 53	54 - 64	65 - 76	77 - 100

Evaluación interna del Nivel Superior – Dossier de trabajo personal

Bandas de calificación del componente

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 5	6 - 10	11 - 17	18 - 25	26 - 33	34 - 41	42 - 50

Ámbito y adecuación del trabajo entregado

La selección de problemas del Nivel Superior es, obligatoriamente, más difícil y se debería tener en cuenta que todos los aspectos de dominio deben planificarse en la fase inicial del proceso. Casi todos los alumnos eligen un problema orientado a una base de datos estándar. Sólo unos cuantos intentaron algo original y, lamentablemente, encontraron problemas. No obstante, en este punto, los aspectos de dominio extendidos para 2010 deberían permitir una mayor flexibilidad.

La mayoría de alumnos comprendió la importancia de los criterios de evaluación y modelaron su trabajo para conseguir dichos criterios correctamente. Parece que muchos profesores enseñaron a sus alumnos un modelo de dossier y les animaron a elaborar uno similar. No obstante, habría que tener cuidado a la hora de realizar una aproximación excesivamente basada en un modelo que hiciera que los alumnos no reflexionaran con profundidad sobre los problemas y que condujera a la aparición de plagios y colaboraciones. La línea que separa estos conceptos es muy sutil.

Siguen sin comprender correctamente los conceptos de programación orientada a objetos (POO) en muchos casos y, a menudo, se usan de forma trivial. Por ejemplo, el uso de la palabra clave `extends` no demuestra el dominio de la herencia. Cuando un alumno emplea la herencia, debería tener planificado su uso y desarrollar todas las clases necesarias. La única excepción se produce cuando se extiende considerablemente una biblioteca de clases existente y el alumno puede explicar por qué elige esa aproximación.

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

En la mayoría de ocasiones se considera trivial ofrecer un constructor vacío y luego otro para inicializar los datos miembros de una clase. El alumno tiene que justificar por qué usa varios constructores para demostrar que la solución se beneficia de ese diseño.

Analizar sintácticamente un fichero de texto u otro flujo de datos significa algo más que usar un método o dos de las clases incorporadas (e.g. `Integer.parseInt(String)` y `Double.parseDouble(String)`). Esto es sólo un proceso de conversión de un tipo de datos a otro. Este concepto implica tomar una cadena de texto, dividirla o separar sus identificadores en función de los caracteres del flujo para generar los datos que se van a procesar posteriormente.

Se sigue usando la recursividad en situaciones en las que un simple bucle iterativo podría funcionar e incluso ser más adecuado. Esto también se considera trivial.

Añadir datos a una instancia de `RandomAccessFile` usando el método `seek`:

```
miFichero.seek(miFichero.length())
```

también es un ejemplo trivial de cómo añadir datos a una instancia de la clase `RandomAccessFile`.

Se puede encontrar más información sobre los aspectos de dominio individual en la tabla que se muestra al final de las recomendaciones para futuros alumnos, así como en los informes generales de la asignatura (mayo de 2006) disponible en el centro pedagógico en línea.

Desempeño de los alumnos con relación a cada criterio

Suelen obtener notas similares en todos los criterios. Es decir, los alumnos menos preparados obtienen notas bajas en muchos criterios, mientras que los más preparados consiguen notas más altas en la mayoría de los criterios.

Los criterios en que se obtuvieron por lo general los mejores resultados fueron: A3, C1, C2, C3, D2 y D3. En los otros criterios hubo una gran variedad de problemas.

Criterio A1

En A1, muchos alumnos comienzan con una visión global, no específica, del problema y posteriormente pasan a una discusión sobre su solución. Debería haber un desarrollo gradual de los detalles del problema que conduzca de forma lógica a una solución

propuesta. A menudo, se obvia el contexto, presentando el problema de forma confusa al lector (en cuanto se refiere a los dossiers del NM).

Criterio A2

Muchos alumnos escribieron una lista coherente de objetivos. En muchos casos, la lista de objetivos era tan larga que habría sido virtualmente imposible ofrecer ejemplos para demostrar el correcto funcionamiento del programa. Además, muchos de los objetivos están detallados innecesariamente o requieren detalles técnicos sin interés, por ejemplo: “el programa debería almacenar los datos en un RandomAccessFile usando 100 bytes por registro”. Los criterios deben ser específicos y perceptibles, pero los detalles técnicos no.

Criterio A3

Algunos alumnos omiten o malinterpretan el requisito para un diseño inicial, que puede ser un simple diagrama del módulo surgido de las secciones A1 y A2. Las evidencias sobre los comentarios del usuario a menudo están incompletas y son poco convincentes.

Criterio B1

Las estructuras de datos deberían explicarse en función de los requisitos de la solución propuesta. Muchos alumnos esbozan una lista enlazada genérica pero sin datos de ejemplo o explicación alguna sobre cómo debe usarse. Por tanto, esta documentación no proporciona ninguna información que ayude al lector a entender el diseño global de la solución.

Criterio B2

Este requisito se reducirá notablemente a partir de 2010. De esta forma, se debería reducir la necesidad de que los alumnos resuman (o incluso copien) el código Java final. Sólo realizan correctamente esta sección en contadas ocasiones.

Criterio B3

A menudo los alumnos se limitan a presentar una descripción de las clases que ya han escrito y los profesores, por desgracia, creen que esto vale 4 puntos. El diagrama de un módulo contribuiría en gran medida a cumplir este criterio, especialmente si se incluyen las conexiones con las estructuras de datos y los algoritmos.

Criterio C1

Los listados de código son, por lo general, correctos. Los alumnos no siempre delimitan el código generado por el IDE del código escrito por ellos mismos.

Criterios C2 y C3

Hay que tener en cuenta que la sección de Facilidad de uso no será necesaria a partir de mayo de 2010.

En la sección sobre el tratamiento de errores se debería hacer referencia a ejemplos de código, citando código o indicando métodos y líneas concretas.

Criterio C4

Para esta sección no se requiere documentación específica.

DOCUMENTACIÓN

Los alumnos no deberían obviar esta sección, ya que para conseguir los aspectos de dominio y los criterios se requiere un número considerable de pruebas.

Criterio D1

Es importante que los alumnos del NS usen esta sección para demostrar el dominio de los aspectos exigidos, cuando proceda. Los profesores deberían hacer un esfuerzo para asegurarse de que todos los alumnos ofrecen al menos un ejemplo de ejecución completa de la solución; aunque, según la guía, sean necesarias más de una. No obstante, hay que centrarse menos en los datos incorrectos.

Criterio D2

En el NM no siempre realizan correctamente esta sección. En este nivel debería discutirse la eficiencia, y las sugerencias para futuras modificaciones deberían ser prácticas y realistas, es decir, dentro de las capacidades del proponente.

Criterio D3

Esta sección no será necesaria a partir de 2010.

Recomendaciones para la enseñanza a futuros alumnos

Los problemas de las secciones A1, A2, B1 y B2 fueron, a menudo, el resultado de que los alumnos escribieran primero el programa.

Los profesores deberían hacer hincapié en la importancia de ofrecer datos de ejemplo para el ANÁLISIS y las PRUEBAS de un sistema informático. Aunque ya no se requiere un juego de pruebas riguroso, los alumnos tienen que comprender todavía que al usuario le importa mucho más los datos que el software. El objetivo de la mayoría de los programas de usuario es la correcta gestión de los datos.

Los profesores deberían insistir en que los alumnos realizaran primero el análisis y posteriormente el diseño antes de comenzar a programar. Posteriormente, los profesores deben ofrecer orientaciones cuando los alumnos sigan el método incorrecto. Los profesores tienen que saber cuándo un alumno está pensando en una solución inadecuada (demasiado simple o demasiado compleja), advertirle y ayudarlo a corregir su razonamiento en las fases iniciales del proyecto. Esto ahorra tiempo y esfuerzo y aumenta notablemente la probabilidad de que el alumno tenga éxito.

Casi todos los alumnos han adoptado la estrategia segura de elegir un problema orientado a las bases de datos implementado en un programa de consola. Se espera que los cambios en los criterios de evaluación –especialmente en el NS- animarán a los alumnos a aventurarse un poco y a que prueben otros tipos de problemas.

La información sobre los aspectos individuales de dominio de la tabla siguiente debería leerse junto con los informes generales de la asignatura de mayo de 2006 disponible en el CPEL.

ASPECTO DE DOMINIO DE NS	QUÉ ES ACEPTABLE	LO QUE NO SE DEBE HACER
Añadir datos a una instancia de la clase RandomAccessFile	El método seek debería usarse sólo para mover el puntero del fichero a un byte específico del fichero antes de escribir.	<ul style="list-style-type: none"> • buscar desde el principio del fichero – seek(0) – y escribir un flujo de datos en el fichero. • buscar hasta el final del fichero (fichero.length) y añadir los datos al fichero.
Eliminar datos de una instancia de la clase RandomAccessFile	El método seek debería usarse sólo para mover el puntero del fichero a un byte específico del fichero antes de leer.	Buscar desde el inicio del fichero (seek(0)) y posteriormente leer el fichero completo es un uso trivial.
Buscar datos concretos en una instancia de la clase RandomAccessFile	No es necesaria ninguna aclaración	No es necesaria ninguna aclaración
Recursividad	No es necesaria ninguna aclaración	<ul style="list-style-type: none"> • Se considera un uso trivial de la recursividad aquél en que una solución iterativa funcionaría igualmente (p. ej. contar nodos o buscar en una lista enlazada). • No se debería argumentar el uso de la recursividad cuando se realice un algoritmo de búsqueda estándar (por fusión u ordenación rápida) que no se haya

		documentado en la fase de diseño.
Fusionar dos o más estructuras de datos ordenadas	Fusionar requiere dos conjuntos de datos que ya están ordenados.	El error más común es realizar una ordenación por fusión.
Polimorfismo	No es necesaria ninguna aclaración	Los alumnos no deberían usar código de biblioteca para exigir el dominio, ya que ellos no han escrito ninguna de esas superclases.
Herencia	El alumno debe documentar y mostrar cómo la subclase usa los métodos y los miembros dato de la clase padre.	La herencia no puede extender sólo a una clase incorporada o de biblioteca.
Encapsulación	No es necesaria ninguna aclaración	No es necesaria ninguna aclaración
Análisis sintáctico de un fichero de texto u otro flujo de datos	No es necesaria ninguna aclaración	El uso de clases incorporadas tipo wrapper, como Integer y Double con los métodos <code>parseInt(String)</code> y <code>parseDouble(String)</code> se considera trivial.
Implementación de una estructura de datos jerárquica compuesta	Se pueden encontrar ejemplos en la página 65 de la guía de la asignatura.	No es necesaria ninguna aclaración
Uso de alguno de los cinco factores de dominio del nivel medio	Consultar la tabla de puntuaciones de los factores de dominio del NM	No es necesaria ninguna aclaración
Se pueden otorgar un máximo de 4 aspectos por la implementación de tipos de datos abstractos (TDA)	Cuando uno o más alumnos usen el mismo TDA el profesor debe realizar una anotación en relación con cada alumno; asimismo, los alumnos deben justificar individualmente el uso de la estructura de datos (que será distinto para diferentes problemas).	No es necesaria ninguna aclaración

Uso de bibliotecas adicionales	El alumno también debe escribir personalmente código que use la biblioteca, y dicho código debería estar identificado claramente como propio.	Reclamar el uso de bibliotecas necesarias para otros aspectos de dominio (como el uso de <code>java.io.*</code> para la gestión de ficheros). No se tiene en cuenta el uso de clases creadas por el alumno.
Inserción de datos en un fichero secuencial ordenado sin leer el fichero completo en memoria RAM.	Se espera que el alumno haga algo más que almacenar datos.	Los alumnos del NS deberían evitar el uso de la biblioteca SQL con una base de datos externa o la serialización.
Eliminación de datos de un fichero secuencia sin leer el fichero completo en memoria RAM.		
Matrices de dos o más dimensiones.	Los alumnos, por tanto, deberían usar matrices estándares con subíndices entre corchetes.	La clase <code>java.io.ArrayList</code> u otras implementaciones similares a los vectores deberían evitarse, ya que este aspecto se refiere a matrices estáticas tradicionales.

Evaluación interna del Nivel Medio – Dossier de trabajo personal

Bandas de calificación del componente

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 6	7 - 13	14 - 20	21 - 27	28 - 35	36 - 42	43 - 50

Ámbito y adecuación del trabajo entregado

La selección de problemas y la presentación de los dossiers en esta sesión fueron buenas o muy buenas.

Aún hay alguna elección incorrecta, entre las cuales se encuentran juegos, problemas basados en juegos o limitados en cierta medida. Algunos proyectos no pudieron conseguir los aspectos de dominio requeridos o no existía un usuario real, lo que evitó que el alumno fuera capaz de abarcar el problema.

Algunos dossiers resultaron demasiado complejos, y esto suele derivar en demasiadas clases con demasiadas variables de instancia que incrementan la carga de trabajo del alumno sin ningún beneficio.

Los objetivos deberían indicarse clara y específicamente y, en la mayoría de los casos, 4 objetivos pueden conducir a un problema trivial y 8 o más a una problema excesivamente complejo. Los alumnos deberían ser capaces de explicar con claridad al profesor (que debe ofrecer una orientación adecuada en esta área) cómo demostrarán su corrección. Si un/a profesor/a, debido a su inexperiencia tiene alguna duda sobre el programa debería pedir consejo en el CPEL.

Los alumnos que eligieron un usuario final demasiado cercano y accesible (como un profesor o un familiar) tuvieron más facilidades para recopilar los datos necesarios para el análisis y el establecimiento de requisitos.

Los alumnos del NM obtuvieron por lo general 10 aspectos de dominio, aunque algunos incluyeron comentarios irrelevantes en sus programas sólo para lograr un elemento de dominio. Por ejemplo, un programa que sólo tiene un método adicional que imprime cadenas de texto por la consola no es suficiente para demostrar el dominio de los métodos. Los alumnos deberían estar familiarizados con los métodos y, por tanto, en un dossier debería haber bastantes métodos.

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 profesores deben asegurarse de que los alumnos entienden qué significa cada factor de dominio y pedir asesoramiento, por ejemplo, en el CPEL si fuera necesario, así como usar la información de la tabla que aparece después de las recomendaciones para la enseñanza de alumnos futuros.

Hay algunas pruebas que demuestran que los profesores no son competentes en esta área y se dedicaron a seguir (indebidamente) las indicaciones del alumno.

La E/S de ficheros implica las acciones de escribir y leer ficheros (los ficheros de texto son válidos), no sólo una de ellas. Las matrices deberían ser de tipos estáticos y no se deberían usar utilidades de biblioteca como ArrayList y Vector para este aspecto.

Aunque los métodos “setter” (accesores) y “getter” (modificadores) y los constructores se usen como parte de un objeto no deberían usarse para exigir el dominio de métodos con parámetros que devuelven valores, ya que esos métodos suelen ser triviales. Además, parece que muchos alumnos usan el especificador de ámbito público (que hace que esos métodos sean superfluos) por defecto en una clase.

Algunos alumnos del NM no han recibido puntos por el uso de indicadores/centinelas o bibliotecas adicionales, cuando sí deberían haberlos obtenido. Se consideran bibliotecas adicionales, por ejemplo: AWT, bibliotecas de interfaces gráficas de usuario como Swing o utilidades como StringTokenizer, ArrayList o LinkedList. El dominio de bibliotecas adicionales no debería, en cambio, exigirse por usar java.io en la gestión de ficheros.

Con frecuencia, los alumnos ofrecen pocos ejemplos de ejecuciones de sus programas. Esto es esencial para que el moderador confirme que se han conseguido los aspectos de dominio, que el programa funciona realmente como se indicó y verificar la evaluación del profesor en C1. Es virtualmente imposible confirmar el dominio de matrices, de E/S en ficheros o de la

búsqueda y clasificación cuando no se ofrece una muestra impresa. El único recurso que tiene el moderador es probar e interpretar la corrección potencial o a partir del listado de código. Muchos moderadores tienen suficiente paciencia para ello, aunque no se espera que lo hagan. Es el alumno el que tiene la responsabilidad de demostrar la corrección de su trabajo.

Los alumnos deberían ser supervisados de cerca en cada fase del dossier, para que el profesor sepa cómo progresa cada uno y pueda verificar (en 5/PDCS, DTP de informática) que el trabajo corresponde en exclusiva al alumno que lo ha realizado competentemente. En este formulario los profesores no deberían escribir anotaciones del tipo "el alumno envió el dossier sólo en la fecha límite y por tanto no puedo confirmar que es trabajo suyo".

Desempeño de los alumnos con relación a cada criterio

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 hace ingeniería inversa en el análisis suelen obtener calificaciones bajas en esta sección y en el resto del dossier.

Criterio A1

Muchos alumnos siguen empezando por hablar del programa que van a desarrollar (o que probablemente ya hayan escrito) en lugar del problema de información que están investigando. Muchos alumnos asumen el conocimiento del dominio del problema y se les debería animar a que ofrecieran una introducción general.

Debería haber pruebas de la recopilación de datos. Esto implica presentar datos reales: números y texto del dominio del problema. Muchos dossiers contienen explicaciones, por ejemplo, "nombre y número de teléfono", pero deberían mostrarse también datos de prueba, como por ejemplo "Fred Thomas, 312-4567". Los alumnos que ofrecieron estos datos casi siempre obtuvieron mejores puntuaciones en su dossier que aquellos cuyo usuario final era ficticio o parcialmente inventado.

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 alumnos de informática deberían ser expertos en el uso de escáneres y cámaras digitales para capturar e incluir datos relevantes: la mayoría del resto de alumnos ya lo son.

En contadas ocasiones se realizó un análisis sistemático. En un análisis sistemático, se describen de alguna manera los procesos empleados para producir las salidas necesarias a partir de las entradas cuidadosamente seleccionadas. Esto se puede realizar con una descripción o un diagrama (historias de usuarios, casos de uso, diagramas de flujo de datos, etc.). Como todos los diagramas del dossier, deberían estar relacionados con el problema particular en lugar de ser ejemplos genéricos.

Criterio A2

Los alumnos deberían relacionar explícitamente los objetivos con el análisis. Esto implica algún tipo de explicación acerca de la importancia de cada objetivo. Por ejemplo:

"Se debería poder buscar por color o número de puertas en la lista de coches, ya que los usuarios suelen tener requisitos específicos cuando se compran un coche usado"

Como se indica anteriormente, unos 6 objetos de este tipo pueden crear un proyecto de un ámbito razonable. Los objetivos deben adecuarse al plan SMART (por sus iniciales en inglés Specific, Measurable, Achievable, Relevant y Time-constrained, i.e., específico, medible, factible, relevante y restringido en el tiempo). Como mínimo deben ser específicos y medibles, para que el alumno sepa cuándo los han completado (en C4), que puede demostrarlos (D1) y discutirlos (D3).

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

Algunos alumnos 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.

Para este criterio, algunos alumnos se limitan a incluir sus capturas de pantalla de la solución. El uso de prototipos resulta una habilidad útil transferible, de modo que es una oportunidad perdida.

DISEÑO

La sección de diseño sigue ocasionando a los alumnos la mayoría de problemas y es debido, normalmente, a que no se ha realizado un análisis exhaustivo del problema.

Criterio B1

Los objetos de las interfaces gráficas de usuario como cuadros de texto, listas desplegadas y similares no se consideran relevantes para esta sección. Los alumnos del NM deben centrarse principalmente en los tipos de datos que hayan usado o en cualquier estructura (fundamentalmente matrices y ficheros) que estén pensando usar. Si se usa un objeto similar a un registro también se debería incluir la descripción del mismo.

Las estructuras de datos deberían discutirse e ilustrarse con bocetos y ejemplos del dominio. Los alumnos peor preparados no realizaron correctamente esta sección y fueron incapaces de ofrecer estos elementos.

Los alumnos deberían evitar ofrecer explicaciones extensas sobre las estructuras de datos empleadas y usar diagramas para aclarar y comunicar eficazmente sus ideas. Ésta suele ser una buena oportunidad para que unos alumnos revisen el trabajo de otros. Se debería animar a los alumnos a que realizaran el diseño en la fase inicial de esta sección.

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.

Esto ya no será un problema a partir de 2010, ya que no se requerirán descripciones detalladas de algoritmos.

Criterio B3

Para la mayoría de alumnos sería suficiente un conjunto relativamente simple de módulos que dejaran claras las conexiones entre los algoritmos y las estructuras de datos. En caso de duda, para la mayoría de los alumnos un diagrama suele ser una aproximación más simple y mejor que una aproximación en papel.

EL PROGRAMA**Criterio C1**

Se debería usar un sistema claro y coherente de numeración de páginas. Los profesores deberían comprobar que los aspectos de dominio están correctamente referenciados. Los listados de códigos suelen ser correctos, aunque en ocasiones el uso de tipos de letra extraños y espaciados de línea dificultan la lectura. Un tipo de letra con espaciado simple es la mejor opción.

Criterios C2 y C3

En la sección de facilidad de uso, los alumnos que hacen referencia explícita a los criterios de facilidad de uso 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 facilidad de uso.

Hay que tener en cuenta que la sección de Facilidad de uso no será necesaria en mayo de 2010.

En la sección sobre el tratamiento de errores se debería hacer referencia a ejemplos de código, citando código o referenciando métodos y líneas concretas.

Criterio C4

Una aproximación sólida que hace pensar al alumno en elaborar una sección D1 eficaz es describir cómo el programa cumple cada objetivo, incluyendo salida impresa que demuestre que el programa funciona realmente como se espera. No obstante, no es un requisito para el dossier, sólo una sugerencia que ayuda al alumno a que evalúe y critique el trabajo que ha realizado.

DOCUMENTACIÓN

Los alumnos no deberían obviar esta sección, ya que para conseguir los aspectos de dominio y los criterios se requiere un número considerable de pruebas.

Criterio D1

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. En la guía se afirma que una sola ejecución con datos válidos casi nunca es suficiente. Se deberían realizar más pruebas con datos correctos que con datos incorrectos. Es mucho mejor que el alumno se centre en lo que ha conseguido. Las tablas de ejemplos propuestas no son imprescindibles, pero los alumnos que las siguieron obtuvieron mejores puntuaciones.

Las pruebas con resultados incorrectos no se debería pasar por alto, aunque quizá debería constituir el 30% de las capturas de pantalla en lugar del 90%.

Para confirmar la concesión de estos aspectos de dominio es imprescindible aportar pruebas de que se pueden añadir varios registros a las matrices y los ficheros.

Criterio D2

Con frecuencia, los alumnos se muestran reacios a mencionar las partes de su solución que no funcionan o no lo hacen correctamente, pensando en que probablemente los examinadores no se darán cuenta. Algunas veces se centran más en las posibles mejoras, en detrimento de la solución real.

Muchos alumnos –y con razón- están orgullosos de sus logros, pero ésta no debería ser la única consideración en una evaluación.

Criterio D3

Esta sección no será necesaria a partir de 2010.

Recomendaciones para la enseñanza a futuros alumnos

El problema más común de esta sección fue la falta de ejecuciones de ejemplo con datos válidos. Los alumnos y profesores deberían tener en cuenta que éste es el único modo que tienen los examinadores de evaluar los logros reales del alumno.

Como se ha indicado, hay mucho trabajo por hacer antes de iniciar el proyecto (y aún más antes de empezar la codificación). Los profesores deben, junto con los alumnos, comprobar la naturaleza del proyecto y ver si se van a cumplir los aspectos de dominio y si estos van a ser suficientes. También deben asegurarse de que los alumnos entienden los criterios. Cuando los profesores tengan dudas sobre los aspectos de dominio deben resolverlas en la fase inicial del proceso.

Una forma de resolver la duda inicial es hacerse la pregunta “¿qué debería ser capaz de hacer una solución para este problema?” para guiar el proceso a través de los criterios hasta el éxito. La mejor forma de realizar correctamente el análisis en A1-A3 es tener un usuario real.

En algunos casos se intentó afrontar problemas excesivamente grandes y complejos. Aun cuando se realice correctamente, el alumno tendrá que obviar algunos aspectos de su estudio. El problema debe seguir siendo simple y no debemos pensar que hay que cumplir todos los requisitos del usuario. Una vez más, los profesores tienen un papel importante a la

hora de orientar a los alumnos. Un buen dossier del NM, para los nuevos criterios de 2010, se puede conseguir con unas 40 páginas.

La información sobre los aspectos individuales de dominio de la tabla siguiente debería leerse junto con los informes de mayo de 2006 disponible en el CPEL.

ASPECTO DE DOMINIO DE NM	QUÉ ES ACEPTABLE	LO QUE NO SE DEBE HACER
Matrices	Implementar una matriz estándar de tipos primitivos en Java como <code>int [] puntuacion = new int[10]</code> o de tipo objeto como <code>. nombreObjeto [] lista = new nombreObjeto[10]</code> .	NO SE DEBE USAR ArrayList o cualquier otra lista o colección como HashMap, etc.
Objetos definidos por el usuario	El uso de los objetos debe ser único para la solución al problema y debe haber sido creado por el programador; p. ej. un objeto círculo con un radio.	Exigir clases incorporadas de Java, p. ej. Swing. Algunas clases como los manejadores Exception se pueden usar siempre que el alumno añada un nivel de complejidad razonable y justifique su uso.
Objetos como registros de datos	Normalmente, un objeto con datos asociados y métodos usados para almacenar una entidad cuyos datos forman un registro en un fichero.	NO se debe usar ninguna clase incorporada de Java.
Selección simple	No es necesaria ninguna aclaración	NO se debe usar ningún código que no haya escrito el alumno.
Selección compleja	No es necesaria ninguna aclaración	NO se debe usar ningún código que no haya escrito el alumno.
Bucles	No es necesaria ninguna aclaración	NO se debe usar ningún código que no haya escrito el alumno.
Bucles anidados	No es necesaria ninguna aclaración	NO se debe usar ningún código que no haya escrito el alumno.
Métodos definidos por el usuario	No es necesaria ninguna aclaración	NO se debe usar ningún código que no haya escrito el alumno.

Métodos y parámetros definidos por el usuario	No es necesaria ninguna aclaración	NO se debe usar ningún código que no haya escrito el alumno.
Métodos definidos por el usuario que devuelven valores	No es necesaria ninguna aclaración	NO se debe usar ningún código que no haya escrito el alumno.
Ordenación	Código escrito por el alumno para realizar una ordenación por el método de la burbuja o de inserción. Consultar el material de apoyo al profesor en el CPEL para obtener ejemplos.	NO se deben usar métodos de ordenación incorporados en Java asociados a una clase, como Collections o <code>Java.util.Arrays.sort(nombreMatriz)</code>
Búsqueda	Código escrito por el alumno para realizar una búsqueda lineal o usando el método de la burbuja. Consultar el material de apoyo al profesor en el CPEL para obtener ejemplos.	NO se deben usar métodos de búsqueda incorporados en Java asociados a una clase, como Collections o <code>BinarySearch()</code>
E/S de ficheros	Operaciones de lectura y escritura sobre ficheros secuencias usando <code>BUfferedReader</code> y <code>PrintWriter</code> como se esboza en la guía de la asignatura o clases similares P. ej. <code>FileOutPutStream</code> o <code>DataInputStream</code> . La serialización de objetos se puede usar en el NM para leer y escribir datos desde/hacia almacenamientos permanentes, aunque debe explicarse correctamente y justificarse. El uso de conexiones Java con bases de datos	Hay que evitar el uso de la serialización de objetos con clases Java como <code>ArrayList</code> o <code>HasMaps</code> si se pretende conseguir otros factores de dominio relacionados con la búsqueda y la ordenación.

	externas como SQL se permite para la E/S de ficheros, pero no para reclamar el dominio de la ordenación y la búsqueda.	
Uso de bibliotecas adicionales	Se pueden reclamar para cualquier uso válido de una clase Java asociada a la resolución del problema.	NO se debe reclamar el uso de clases que no se empleen para resolver el problema.
Uso de centinelas o indicadores	Los centinelas se pueden usar para indicar el final de un fichero secuencial, el final de los datos en una matriz o al introducir datos desde un teclado. Se puede implementar un simple indicador para finalizar una ordenación por el método de la burbuja o una búsqueda lineal.	NO se debe reclamar un centinela o un indicador implementado en una clase Java que no haya escrito el alumno.

Prueba 1 del Nivel Superior

Bandas de calificación del componente

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 15	16 - 30	31 - 41	42 - 51	52 - 62	63 - 72	73 - 100

Áreas del programa y del examen que resultaron difíciles para los alumnos

Muchos alumnos lo hicieron razonablemente bien. Los colegios deben recalcar a sus alumnos que ofrezcan respuestas precisas y concretas.

Puede que la mayoría de profesores y alumnos no hayan reflexionado sobre el problema de interfaces (pregunta 18). Sólo los alumnos muy preparados respondieron correctamente a esta pregunta.

Áreas del programa y del examen en que los alumnos demostraron estar bien preparados

El nivel de conocimientos de los alumnos que se presentan a este examen es bastante variado. Parece que la mayoría de colegios ha cubierto el programa de estudios y el

rendimiento de muchos alumnos se sitúa por encima de la media. Hay unos cuantos alumnos que obtienen resultados extremadamente buenos y otros con resultados paupérrimos.

Muchos alumnos ofrecen respuestas muy breves y que no tienen en cuenta el número de puntos asignado a cada pregunta.

Puntos fuertes y débiles de los alumnos al abordar las distintas preguntas

SECCIÓN A

Todas las preguntas de esta sección parecen asequibles para los alumnos y la mayoría muestran una buena cobertura del programa de estudios.

Pregunta 1

Muchos alumnos respondieron correctamente. Unos cuantos alumnos afirmaron que “la memoria RAM y la ROM se almacenan en el computador”. Necesitan prestar más atención a la terminología.

Pregunta 2

Muchos alumnos no encontraron dificultades en la pregunta sobre sistemas operativos. Unos cuantos ofrecieron explicaciones muy rebuscadas sobre las funciones de un sistema operativo, aunque no era lo que se esperaba.

Pregunta 3

La mayoría de alumnos respondió con seguridad la pregunta sobre representación de números de 5 bits en complemento a dos.

Pregunta 4

Sobre las ventajas de la “compresión de datos” se ofrecieron muy buenas respuestas, como menos espacio en disco (en realidad, más datos) y una mayor rapidez en lectura, escritura y transferencia de ficheros.

Pregunta 5

Sorprendentemente, muchos alumnos respondieron “un compilador convierte lenguaje de alto nivel en código máquina”, en lugar de “un compilador convierte código fuente escrito en un lenguaje de programación en código máquina”.

Pregunta 6

La mayoría supo describir con seguridad la diferencia entre un error lógico y un error sintáctico.

Pregunta 7

Sorprendentemente, muchos alumnos respondieron que los métodos definidos por el usuario son diseñados por el usuario final y no por el programador.

Pregunta 8

Entre las muchas respuestas correctas se incluyen “un menor tiempo de desarrollo”, “participan varios equipos de desarrollo” o “reduce el número de errores.

Pregunta 9

Todos los alumnos respondieron relativamente bien a la pregunta sobre análisis y diseño de sistemas.

Pregunta 10

Responden bien a esta pregunta.

Pregunta 11

La mayoría de alumnos simplificó correctamente la expresión booleana.

Pregunta 12

La notación O Mayúscula (la “o” se refiere a “orden de”) hace referencia a lo que ocurre con valores muy grandes de N y, por tanto sólo hay que fijarse en el término de mayor grado del polinomio. Algunos alumnos poco preparados se limitaron a escribir respuestas imprecisas que no merecían ningún punto.

Pregunta 13

La mayoría de alumnos obtuvieron buenas puntuaciones por definir el significado del término WAN (parte a), aunque sólo unos cuantos ofrecieron una explicación de los beneficios de usar la conmutación de paquetes en una WAN (parte b).

Pregunta 14

Responden bien a esta pregunta.

Pregunta 15

Las variables que sólo existen dentro de un bloque o método se denominan variables locales, pues tienen un alcance local. Muchos alumnos respondieron correctamente.

SECCIÓN B

Pregunta 16

La mayoría de alumnos que intentó responder a esta pregunta lo hizo extremadamente bien o extremadamente mal. Lo cual hace pensar que sus habilidades para programar y su comprensión eran correctas y que pudieron aplicar sus conocimientos a esta pregunta o, por el contrario, que carecían de tales habilidades y conocimientos.

Pregunta 17

Algunos alumnos, al responder a la pregunta relacionada con el teclado que permite acceder a una sala informática, ofrecen explicaciones extensas sobre cómo las operaciones de pila (“push” y “pop”) ayudan al procesamiento.

Pregunta 18

La única dificultad real de esta pregunta era la parte c. La mayoría de alumnos no ha tenido en cuenta el problema de interfaces.

Pregunta 19

Algunos de los alumnos más inteligentes no tuvieron ninguna dificultad para sugerir la estructura de datos en un escenario concreto. Fueron capaces de justificar sus elecciones de forma convincente. Unas cuantas respuestas incluían diagramas como complemento a la respuesta del alumno.

Algunos de los alumnos tuvieron dificultades para elegir correctamente la estructura de datos necesaria para responder a una pregunta. No supieron identificar ninguna estructura de datos, aunque la pregunta era muy específica en cuanto al nombre de la estructura de datos correcta. Se esperaba que los alumnos respaldaran sus elecciones explicando por qué habían seleccionado una estructura de datos concreta.

Pregunta 20

Algunos alumnos tienen dificultades para comprender que la **herencia** es la posibilidad que tiene una clase de usar las propiedades y métodos de otra clase a la vez que añaden su propia funcionalidad. Y que el **polimorfismo** es la capacidad que tiene una acción o **método** de realizar operaciones diferentes en función del objeto que las invoca. Estos son los principios básicos de la programación orientada a objetos. Si los alumnos hubieran aprendido estas bases correctamente, esta pregunta no había presentado problemas para ellos.

Pregunta 21

Prácticamente todos los alumnos construyeron correctamente la tabla de verdad de la parte (a). Muchos diagramas usaron diagramas K para minimizar la expresión booleana. La respuesta a la parte c dependía de la expresión simplificada dada en la respuesta a la parte b. Unos cuantos alumnos no supieron dibujar un circuito para la expresión.

Recomendaciones y orientación para la enseñanza a futuros alumnos

Cuanto más exámenes practiquen los alumnos, mejor. Esto no implica, sin embargo, que realicen exámenes continuamente a lo largo del curso, sino ejemplos de preguntas anteriores que se puedan integrar en el proceso de enseñanza y aprendizaje. Esto ayudará a reforzar los conocimientos y a que se familiaricen con la estructura de las preguntas y con la forma de preguntar en los exámenes. También ayudaría proporcionar una copia del programa de estudios a cada alumno al principio del curso, para que puedan preguntar sobre temas que les resulten difíciles.

Se debería prestar especial atención a los verbos de acción que se usan en las preguntas, así como evitar respuestas generales y centrarse en responder concretamente la pregunta que se hace.

Prueba 2 del Nivel Superior

Bandas de calificación del componente

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 17	18 - 34	35 - 41	42 - 51	52 - 61	62 - 71	72 - 100

Áreas del programa y del examen que resultaron difíciles para los alumnos

Todas las preguntas resultaron accesibles para la mayoría de alumnos, y hay pocos indicios que muestren que la falta de tiempo resultó ser un problema importante en este examen. Las mejores puntuaciones estuvieron ligeramente por encima de 90.

La parte d de la pregunta 2 sobre recursividad fue la más difícil, como se preveía, y supuso un desafío para los mejores alumnos. El tratamiento de ficheros también ocasionó dificultades (en algunos colegios más que en otros).

Áreas del programa y del examen en que los alumnos demostraron estar bien preparados

La mayoría de alumnos demostró una comprensión razonable de los temas evaluados. En los últimos años han mejorado notablemente en la manipulación de matrices y el uso de técnicas de POO. Por lo general, los alumnos demuestran un buen conocimiento global de la mayoría de los temas del estudio de caso.

Puntos fuertes y débiles de los alumnos al abordar las distintas preguntas

Pregunta 1

Esta pregunta se basaba en una matriz de objetos sobre los que se realizaban dos búsquedas estándares: secuencial y binaria.

El éxito de los alumnos a la hora de tratar los dos algoritmos subraya la mejora que han experimentado los colegios en la preparación de sus alumnos para estos tipos de preguntas. Muchos alumnos obtuvieron la máxima puntuación en la búsqueda secuencial. Ahora los alumnos parecen cómodos trabajando con matrices. El hecho de que el primer elemento tenga índice 0 ya no ocasiona problemas a la hora de determinar las condiciones del bucle. Aunque no se restaban puntos por la falta de eficiencia, la mayoría finalizaba la iteración en cuanto se encontraba el objetivo.

La búsqueda binaria resultó ser ligeramente más difícil. Entre algunos errores de poca importancia se encuentra declarar incorrectamente el nuevo valor máximo y el mínimo.

Es destacable que la mayoría de alumnos se sientan cómodos manipulando objetos, algo que ha mejorado considerablemente gracias a que los profesores han incluido en el programa de estudios las principales características de la POO.

Pregunta 2

Acerca de la pregunta 1 que trataba la teoría del NM, el interés estaba en el programa del NS, concretamente en la búsqueda sobre árboles binarios.

En las partes (a) y (c) la mayoría de alumnos pudo obtener puntos, demostrando que entendieron los fundamentos de la construcción y del recorrido de un árbol.

El tratamiento del recorrido a nivel algorítmico resultó más difícil, tal como se demostró en la pregunta sobre recuento de nodos de la parte (d).

En esta pregunta se pedía recorrer el árbol de forma recursiva, devolviendo 1 en el nivel anterior cada vez que la nueva raíz era 'distinta de nulo' o incrementando un contador. La recursividad es un tema difícil para la mayoría de los alumnos, pero incluso así hubo varios alumnos que obtuvieron la máxima puntuación. Se otorgaron puntos a aquellos alumnos que comprendieron los elementos básicos de un algoritmo recursivo: dos casos necesarios, uno como condición de finalización (*if raíz = null*) y otro que llama al método con un parámetro modificado (*nuevo valor de raíz*).

La parte (e) pedía que los alumnos introdujeran un nodo en la posición correcta del árbol.

La mayoría hizo un intento razonable en esta pregunta, pero muchos omitieron el caso de un árbol vacío. Algunos alumnos confundieron la diferencia entre una referencia a un nodo y los miembros dato del nodo y, por tanto, hicieron comparaciones imposibles. Otros no supieron usar un nodo temporal para recorrer el árbol. Unos cuantos alumnos ofrecieron, inesperadamente, una respuesta recursiva, pero el código reducido resultante mostró lo elegante que pueden ser este tipo de soluciones.

Pregunta 3

La pregunta sobre gestión de ficheros resultó ser la pregunta más difícil de la prueba, indicando tal vez que algunos colegios son incapaces de dedicar el tiempo adecuado a este tema en la parte final del curso.

Una vez que se ha creado la tabla, se puede usar un identificador numérico para acceder directamente al registro en disco. La posición del registro está determinada por la aplicación del mismo algoritmo usado en la parte (d).

La mayoría sabía qué es un campo clave y cómo encontrar la clave del algoritmo hash. Los dos problemas principales fueron determinar las estructuras de datos que se podrían usar (la matriz de la parte (b) y especialmente las estructuras alternativas de la parte (f), cuando las respuestas previsibles estaban relacionadas con los ficheros indexados) y comprender por qué se usaba una tabla hash. Muchos alumnos pensaban que todo el fichero estaba en la dirección de memoria principal indicada por el algoritmo hash y no se dieron cuenta de que la tabla hash apuntaba a la ubicación del registro en disco.

Aunque los alumnos hicieron referencia a distintas formas de resolver colisiones (liberar espacio, encadenamiento, o pruebas) no todos indicaron claramente que los pasos adicionales requieren un proceso secuencial hasta identificar el elemento correcto.

Pregunta 4

Los alumnos respondieron bien a la pregunta sobre el estudio de caso y para algunos resultó su salvación. A través de las respuestas ofrecidas por los alumnos (algunas de las cuales

resultaron bastante innovadoras), comprendieron bastante bien que otorgar igualdad de oportunidades a todos los miembros de la sociedad es un tema importante. Los alumnos deberían tener en cuenta siempre el número de puntos asignados a cada pregunta. La parte que ocasionó más problemas fue la (g), sobre avances sociales y económicos, a la que sólo respondieron bien aquellos que, presumiblemente, habían dedicado tiempo a discutir los aspectos del estudio de caso en detalle.

Recomendaciones y orientación para la enseñanza a futuros alumnos

- Tratar con detalle suficiente la gestión de ficheros.
- Los algoritmos se deberían estudiar en todas las partes del programa de estudio.
- Si un algoritmo devuelve un valor en su signatura, se debe devolver un valor en lugar de enviarlo a la salida.
- De igual forma, se deben usar parámetros para pasar datos a un método en lugar de introducirlos por teclado.
- Se debería prestar especial atención a los verbos de acción que se usan en las preguntas, así como evitar respuestas generales y centrarse en responder concretamente la pregunta que se hace.
- Familiarizarse con todos los escenarios y la terminología que se ofrecen en el estudio de caso.

Prueba 1 del Nivel Medio

Bandas de calificación del componente

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 11	12 - 23	24 - 30	31 - 37	38 - 43	44 - 50	51 - 70

Áreas del programa y del examen que resultaron difíciles para los alumnos

Muchos alumnos lo hicieron razonablemente bien. Los colegios deben insistir a sus alumnos que ofrezcan respuestas precisas y concretas.

Áreas del programa y del examen en que los alumnos demostraron estar bien preparados

El nivel de conocimientos de los alumnos que se presentan a este examen es bastante variado. Parece que la mayoría de colegios ha cubierto el programa de estudios y el rendimiento de muchos alumnos se sitúa por encima de la media. Hay unos cuantos alumnos que obtienen resultados extremadamente buenos y otros con resultados paupérrimos.

Muchos alumnos ofrecen respuestas muy breves y que no tienen en cuenta el número de puntos asignado a cada pregunta.

Puntos fuertes y débiles de los alumnos al abordar las distintas preguntas

SECCIÓN A

Todas las preguntas de esta sección parecen asequibles para los alumnos y la mayoría muestran una buena cobertura del programa de estudios.

Pregunta 1

Muchos alumnos se limitaron a enumerar las características de las fases de análisis y diseño, pero obviaron describir las diferencias entre estas dos fases. Hay que tener en cuenta que el verbo de acción requiere no sólo una lista, sino una explicación.

Pregunta 2

Muchos alumnos se limitaron a ofrecer una lista de ventajas en lugar de explicarlas. Otros tuvieron en cuenta sólo las ventajas de un diseño modular y, por tanto, no observaron las ventajas en la construcción, pruebas y mantenimiento de programas informáticos.

Pregunta 3

Muchos alumnos fueron incapaces de obtener la máxima puntuación debido a las respuestas incompletas y poco precisas. La pregunta requería una descripción de cada parte. En dicha descripción había que identificar un sistema o aplicación informática concretos y su efecto en un grupo de individuos bien identificado.

Pregunta 4

La mayoría respondió bien a esta pregunta

Pregunta 5

La mayoría respondió bien a esta pregunta.

Pregunta 6

Algunos alumnos se limitaron a proponer el uso de dos matrices para la parte (b), sin identificar ningún mecanismo para mantener los dos elementos asociados asignados a una persona. Un número sorprendente de alumnos propuso usar una matriz bidimensional para almacenar enteros y booleanos.

Pregunta 7

La mayoría respondió bien a esta pregunta.

Pregunta 8

La mayoría respondió bien a esta pregunta.

Pregunta 9

Muchos alumnos identificaron al usuario final del programa en lugar del programador como la persona que escribe un método definido por el usuario.

Pregunta 10

Por lo general, responden bien a las partes (a) y (b). Algunos alumnos previeron, incorrectamente, que habría cambios en la matriz si hubiera estado ordenada en la parte (c). Muchos alumnos, en la parte (d), no identificaron la necesidad de que el método determinara el tamaño de la matriz en tiempo de ejecución u obviaron sugerir un método para conseguirlo.

Pregunta 11

Muy pocos alumnos respondieron correctamente a esta pregunta. Parece que el uso de un fichero de transacciones para actualizar un fichero maestro dentro de un proceso por lotes resulta desconocido para muchos alumnos. En la parte (a) muchos alumnos no supieron referirse específicamente a los datos que debería haber en un sistema de nóminas y, en su lugar, intentaron escribir descripciones totalmente genéricas de los ficheros maestros y de transacciones. En la parte (c) hubo un número sorprendente de alumnos que identificaron errores accidentales en la entrada de datos en lugar de errores intencionados.

Pregunta 12

En la parte (a) muchos alumnos afirmaron que los ficheros más pequeños se pueden transferir más rápidamente sin que haya diferencias entre la velocidad y el tiempo de transmisión. Los ficheros menores requieren menos tiempo de transferencia debido a que hay menos datos que mover, no porque la velocidad de transferencia sea diferente. En la parte (b) sólo unos pocos alumnos supieron explicar el papel de los protocolos en una red.

Pregunta 13

Esta pregunta se centraba en el uso de un microprocesador incrustado en un dispositivo físico. En cambio, en la parte (b) muchos alumnos identificaron un sistema que era demasiado grande o excesivamente impreciso para ser considerado un dispositivo. En la parte (c) muchos alumnos fueron incapaces de reconocer que cuando un microprocesador se incrusta en un dispositivo, sus entradas y salidas se conectan al hardware del dispositivo y en su lugar describieron las entradas y salidas generadas por el usuario.

Recomendaciones y orientación para la enseñanza a futuros alumnos

Cuanto más exámenes practiquen los alumnos, mejor. Esto no implica, sin embargo, que realicen exámenes continuamente a lo largo del curso, sino ejemplos de preguntas anteriores que se puedan integrar en el proceso de enseñanza y aprendizaje. Esto ayudará a reforzar los conocimientos y a que se familiaricen con la estructura de las preguntas y con la forma de preguntar en los exámenes. También ayudaría proporcionar una copia del programa de estudios a cada alumno al principio del curso, para que puedan preguntar sobre temas que les resulten difíciles.

Se debería prestar especial atención a los verbos de acción que se usan en las preguntas, así como evitar respuestas generales y centrarse en responder concretamente la pregunta que se hace.

Prueba 2 del Nivel Medio

Bandas de calificación del componente

Calificación final:	1	2	3	4	5	6	7
Puntuaciones:	0 - 11	12 - 22	23 - 30	31 - 37	38 - 43	44 - 50	51 - 70

Puntos fuertes y débiles de los alumnos al abordar las distintas preguntas

Pregunta 1

Respondieron razonablemente bien a esta pregunta.

Un número considerable de alumnos realizó un tratamiento correcto de los algoritmos y demostró conocer el paso de valores, la manipulación de matrices, la creación de algoritmos iterativos para acumular totales, la definición de métodos y funciones, la devolución de valores y la gestión de los tipos de datos. No obstante, algunos alumnos demostraron carecer totalmente de habilidades en un área tan importante como es la del desarrollo de algoritmos y programas informáticos.

Pregunta 2

Respondieron razonablemente bien a esta pregunta: casi todos los alumnos saben qué es un fichero secuencial.

Respondieron bien a esta pregunta, y varios alumnos supieron esbozar la lógica básica necesaria. Sin embargo, no está claro que supieran que tenían que describir el fichero.

Parece que en la pregunta (d) los buenos alumnos sabían lo que tenían que hacer, pero otros evidenciaron su falta de trabajo definiendo registros básicos.

Varios alumnos no respondieron bien a la pregunta (e), lo que puede indicar un vacío en la enseñanza de algunos colegios. No obstante, parece que el dossier del NM resulta más fácil si tomamos un problema clásico de gestión de ficheros, lo que requiere manipular un registro a la vez con acceso/organización secuencial y/o leer de una lista y de un fichero ordenado y que es posteriormente actualizado. Por tanto, sigue sorprendiendo que varios alumnos sean incapaces de afrontar este tipo de problema. Muchos respondieron razonablemente bien a la pregunta (f).

El algoritmo resultó familiar a varios alumnos, que disfrutaron desarrollando sus conocimientos ofreciendo descripciones óptimas y extensas de la búsqueda binaria en respuesta a la pregunta (h).

Por desgracia, muchos alumnos no leyeron las palabras "cómo se usa" y por tanto no respondieron correctamente la cuestión.

Pregunta 3

Muchos alumnos obtuvieron casi todos los puntos asignados a esta pregunta.

Las preguntas parecían sencillas y resulta evidente que los alumnos entendieron la importancia del problema abordado en el estudio de caso.

La mayoría de los alumnos demostró una buena capacidad para ofrecer respuestas con sentido.

La última pregunta ocasionó algunos problemas en cuanto a puntuación, ya que muchos alumnos no mencionaron el tipo concreto de documentación, aunque supieron describir una estrategia.

Recomendaciones y orientación para la enseñanza a futuros alumnos

- Los algoritmos se deberían estudiar en todas las partes del programa de estudio.
- Se debería prestar especial atención a los verbos de acción que se usan en las preguntas, así como evitar respuestas generales y centrarse en responder concretamente la pregunta que se hace.
- Familiarizarse con todos los escenarios y la terminología que se ofrecen en el estudio de caso.