

Caso práctico

Víctor ha trabajado duro, y está orgulloso de lo que ha aprendido desde que empezó a formarse como programador en SI Andalucía, y no es para menos. Ahora se da cuenta de la gran diferencia que hay entre programar como afición, con unos cuantos conocimientos aprendidos a base de "torpedear" por su cuenta con compiladores y lenguajes, y programar profesionalmente, con criterios empresariales de calidad y eficiencia. La programación es una tecnología que hay que conocer muy bien para poder sacarle todo el jugo. Por eso algunos libros hablan de ingeniería del software.

José le felicita por todo el esfuerzo de aprendizaje que le ha traído a este punto, y se alegra de poder contar con él para empezar a sacar adelante la programación de alguno de los proyectos en los que están inmersos. Cuenta con él, con sus ganas de aprender y con sus recién adquiridos conocimientos. Y para demostrárselo le ha asignado la programación de una aplicación de pequeño tamaño. Como tienen varios proyectos de aplicaciones pendientes de desarrollar para varios clientes, José le dice a Víctor que elija el que más atractivo le parezca, y lo desarrolle, partiendo de las directrices y especificaciones que se le entregan. También le dice que aunque ha mejorado muchísimo, debe tener presente que un programador nunca termina de formarse, y que siempre va a seguir teniendo bastantes puntos en los que profundizar, tanto en Java como en otros lenguajes y otras filosofías de programación que existen o que irán apareciendo a lo largo de su vida profesional. Por eso, va a contar con la ayuda de Carmen, con la que colaborará trabajando en equipo para desarrollar esa aplicación.

Víctor y Carmen eligen la aplicación a desarrollar de la lista que les suministra José, y se empiezan a planificar el trabajo. A Víctor no le cabe duda de que va a tener que usar todo lo que ha aprendido hasta ahora sobre programación y sobre Java, y tampoco de que deberá enfrentarse a problemas que no había visto hasta ahora. Pero también sabe que parte del trabajo consistirá en seguir investigando aquellas características del lenguaje que le permitan mejorar la calidad final de su aplicación. Pero afronta el reto con optimismo, y se pone a trabajar.

Introducción

Al igual que Víctor, tú también debes aplicar todo lo aprendido a la realización de un proyecto, al desarrollo de una aplicación Java. Se trata de demostrar que has asimilado convenientemente todo lo que has aprendido hasta ahora, y de que lo integres en un único trabajo de complejidad media, para que sea abaricable en unas tres semanas aproximadamente. No es mucho tiempo, incluso para una aplicación de tamaño y complejidad media, por eso tendrás que emplearte a fondo.

En condiciones normales, antes de sentarte a codificar, deberías tener listo un análisis previo del problema que pretendes resolver, y unas especificaciones detalladas del mismo. Aprender a hacer un buen análisis y un buen diseño de la aplicación, completo y detallado, es materia de otro módulo profesional, concretamente de **Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión** (ANA), que tiene tantas horas como este módulo profesional de Programación en Lenguajes Estructurados (PLE) que ahora estás concluyendo. En esta modalidad de formación a distancia es muy posible que no todos los alumnos hayan cursado ese módulo con anterioridad al de PLE, por lo que no podemos ser demasiado exigentes ni rigurosos con este punto. Deberás basarte más en lo que has visto en los numerosos **ejemplos** de programas que se muestran en las unidades anteriores y un poco también en tu intuición y en tu creatividad, aunque creatividad y codificación son conceptos que no debieran estar demasiado mezclados.

A la hora de valorar y evaluar tu trabajo, será más importante la corrección, la implementación de todas las funcionalidades básicas, que disponga de un correcto interfaz gráfico, y la utilidad misma de la aplicación que el hecho de que sea muy vistosa o espectacular.

Consideraciones generales para todos los proyectos

¿Cuando te matriculaste en este módulo tenías en mente alguna aplicación que te gustaría desarrollar? Seguramente alguna vez a lo largo de las unidades de este módulo habrás pensado en alguna aplicación práctica de lo que estabas aprendiendo para uso personal, o para algún conocido o amigo.

Sabemos por experiencia que **los mejores proyectos son siempre los que más motivan a las personas** que los desarrollan. Eso es aplicable a todos los órdenes de la vida, pero especialmente a la

programación. Para que una aplicación sea buena, debe resultar útil, y no hay mejor medida de la utilidad de una aplicación que someterla a los requerimientos de la realidad.

¿Qué queremos decir con esto?

Que a continuación, en los apartados siguientes te vamos a hacer algunas sugerencias sobre posibles **proyectos** que puedes realizar, con el objeto de que elijas uno. Pero esos proyectos, al no ser reales, no pueden ser detallados, ni tienen en consideración detalles que la práctica demostraría imprescindibles en muchos casos.

Por el contrario, si tú mismo te propones desarrollar una aplicación con el objetivo concreto de usarla, sí que vas a tener mucho más claro lo que necesitas que esa aplicación te haga, **y así vas a disponer de unos requerimientos de mucha más calidad que te van a ayudar en la realización del trabajo.**

Además, vas a estar mucho más motivado, porque el trabajo que hagas va a tener un doble fruto:

- Te va a permitir aprobar este módulo profesional, pero al mismo tiempo
- te va a proporcionar una herramienta útil para el propósito que te hayas fijado.

Y si además de cumplir con la tarea de este módulo tu trabajo resulta útil para ti, o para algún conocido tuyo, mucho mejor.

Eso sí, antes de empezar a desarrollar la aplicación, debes ponerte en contacto con tu tutor, e indicarle qué proyecto vas a realizar.

- Si es uno propio que tú mismo te has puesto, deberás **redactar un documento** indicando en qué consiste, y presentárselo a tu tutor para que te dé el visto bueno, o te indique las mejoras que debes tener en cuenta para que cumpla con los requisitos generales que debe cumplir cualquier proyecto.

Pero claro, entre la multitud de ideas que se os pueden ocurrir, no todas van a tener la misma complejidad, ni la misma dificultad, ni el mismo tamaño. Y como queramos o no, tenemos que evaluarlos con unos criterios lo más objetivos y equitativos posible, es necesario que fijemos las características básicas que deberá incluir cualquier proyecto que realicéis, tanto si lo proponéis vosotros mismos como si cogéis uno de los enunciados que mostramos como propuestas en esta unidad.

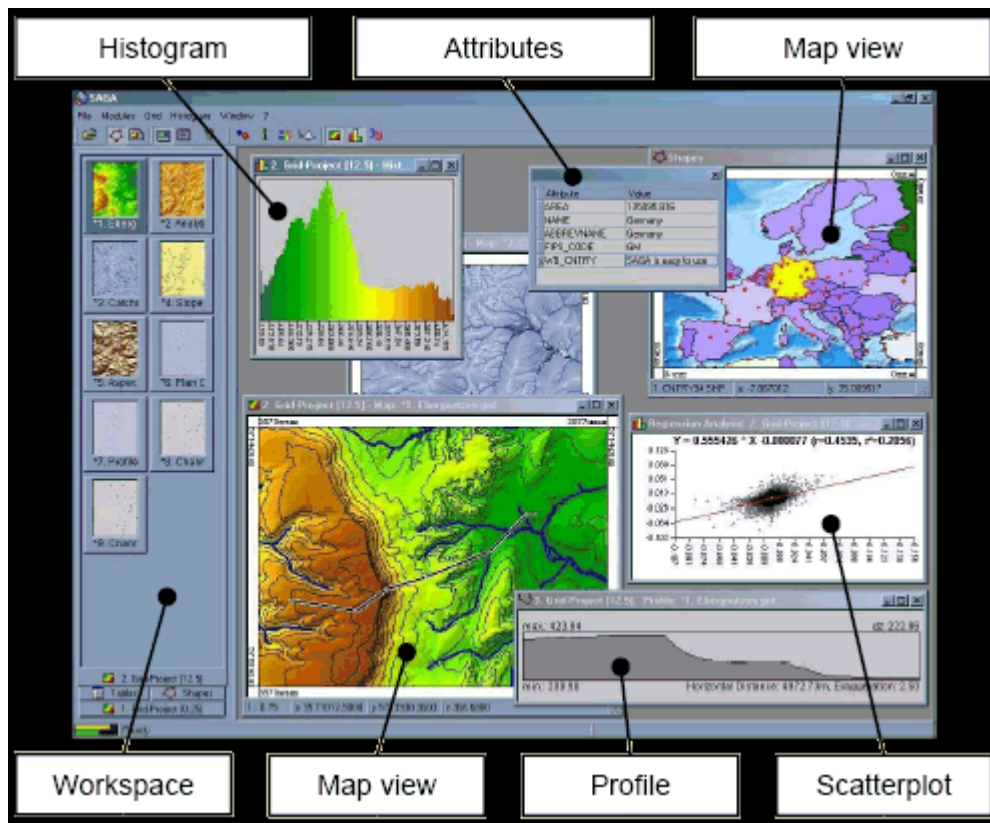
Requisitos mínimos a cumplir por todos los proyectos (I)

¿Pero qué debe hacer tu aplicación, y qué debes tener en cuenta para conseguir una buena calificación de tu proyecto?

El proyecto consiste básicamente en el desarrollo de una aplicación informática por parte de cada uno de los alumnos (de forma individual, o en grupos de dos alumnos. Excepcionalmente, en grupo de tres alumnos) que debe incluir los siguientes puntos:

- Conexión con una Base de Datos situada en un servidor MySQL, (servidor local o servidor remoto, es indiferente). Esta base de datos deberá tener **un mínimo** de 2 tablas de datos.

Diseño de un interfaz gráfico (GUI) apropiado, usando las clases Swing vistas en el curso.
Concretamente, habrá que usar:

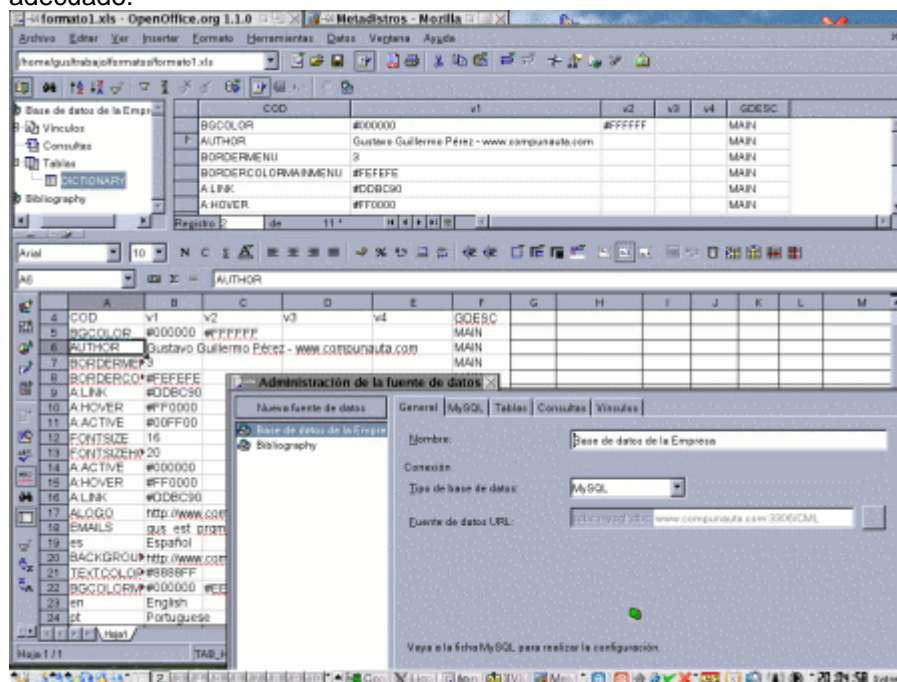


- JFrame para la ventana principal de la aplicación.
- JInternalFrame o JDialog, según convenga, para las distintas ventanas de opciones que se abran dentro de la ventana principal.
- InternalFrameListener, DocumentListener, y algunas clases internas para algunos listener.
- Alguna clase Adapter.
- Tablas JTable para mostrar datos de la BD con formato de tabla (El modelo de datos del JTable debe recoger los datos de la propia BD)
 - Etiquetas JLabel, botones de acción JButton, casillas de verificación JCheckBox, botones de opción o de radio JRadioButton, campos de texto JTextField, etc.
 - Deberás usar al menos un grupo de botones JButtonGroup en el que incluirás varios botones de radio que representen opciones excluyentes, pero de las que debe seleccionarse una obligatoriamente.
 - Paneles de distintos tipos, usados adecuadamente (JDesktopPane, JPanel, JTabbedPane...)
 - Listas JList y/o listas desplegables JComboBox.
 - Por supuesto, deberá incluir una barra de menú JMenuBar adecuada, con distintos menús JMenu y uso de toda la gama de elementos posibles en los mismos (JMenuItem, JCheckBoxMenuItem, JRadioButtonMenuItem, JSeparator), que incluirán sus correspondientes iconos, mnemotécnicos y atajos de teclado al menos en las principales opciones del programa.
 - También deberán usarse distintos tipos de cuadros de diálogo JDialog y JOptionPane (para recoger datos, para informar de algo, para preguntar algo, para notificar errores...)
 - Debe intentarse que el aspecto de la aplicación sea lo más amigable posible, incluyendo ToolTipText, imágenes de fondo en los paneles de las ventanas, iconos, algún gif animado y alguna barra de herramientas (JToolBar) para acceso rápido a las funciones más usadas del menú.





- Aunque por cuestión de premura se use principalmente como gestor de distribución BorderLayout o AbsoluteLayout, que son los que permiten un desarrollo más rápido, debe usarse al menos otro gestor de distribución (Layout) diferente en alguna ventana de la aplicación, de forma que el uso del mismo esté justificado, y suponga una mejora.
- Se valorará positivamente la inclusión del máximo número de componentes Swing de entre los vistos en el curso, siempre y cuando su uso sea oportuno y adecuado.
- Se valorará muy positivamente la inclusión de nuevos componentes Swing de entre los que NO se han visto en el curso, pero de los que el alumno haya investigado su funcionamiento y uso de forma autónoma, siempre y cuando su uso sea oportuno y adecuado.



Requisitos mínimos a cumplir por todos los proyectos (II)

- Mediante el interfaz gráfico debe ser posible:
 - **Consultar y Mostrar (listar)** los datos de la BD. En este punto, **habrá que realizar alguna consulta en la que se vean implicados datos de al menos 2 tablas simultáneamente.**
 - Hacer actualizaciones de los datos de la BD, tanto de **Modificación** de los mismos como de

- **Inserción y Borrado** de elementos en la BD.
 - Disponer de **alguna lista desplegable (JComboBox)** cuyos elementos se obtengan **dinámicamente mediante una consulta a la BD.**
 - Mostrar en alguna tabla los datos contenidos en la BD de forma que pueda **actualizarse el contenido de la tabla** mediante una nueva consulta a la base de datos.
- Deben realizarse **cálculos** de cierta complejidad a partir de los datos de la base de datos, de forma que la aplicación haga algo más con los datos aparte de consultarlos y mantenerlos actualizados mediante las oportunas modificaciones/inserciones/borrados. El objetivo es mostrar el dominio de las estructuras de control de flujo en Java, y el dominio de las técnicas de programación estructurada.
- Deberán **capturarse y tratarse** adecuadamente cuantas excepciones sean necesarias para que el programa no aborte. Cualquier circunstancia de error deberá ser tratada convenientemente.
- Debe **almacenarse** alguno de los resultados obtenidos en ficheros independientes de la BD. Por ejemplo, obtener la suma de las ventas en los distintos meses de cada año, y guardarlo en un fichero "ventas mensuales 2006.dat", o disponer de una opción que guarde en ficheros las quejas y sugerencias de los clientes. (En una aplicación real, no suele ser necesario el uso de ficheros fuera de la BD, pero aquí es necesario que se incluya manejo de ficheros de cara a evaluar esa parte del temario)
- Se pide, de cara a mejorar la nota, que se investiguen e incorporen elementos nuevos, de los disponibles en la API de Java, aunque no hayan sido vistos en el curso, o no se hayan visto con demasiada profundidad, como por ejemplo, la posibilidad de imprimir, o incorporar sonidos, o enlaces a otras aplicaciones externas, como el reproductor de música, o la calculadora, o la generación de algún documento en formato PDF.)
- También para completar la aplicación y mejorar la nota, **debe generarse un recopilatorio** (.jar) para ejecutar la aplicación en cualquier plataforma, sin necesidad del Entorno de Desarrollo (esto justo para entregarlo)
- Debe incluirse la mayor **documentación** posible, incluyendo comentarios dentro del código de forma conveniente, insertando comentarios javadoc, con el propósito de que se genere automáticamente documentación en formato html.
- Debe hacerse un uso adecuado de los conceptos de Programación Orientada a Objetos, al gestionar métodos, interfaces, clases (incluyendo abstractas, internas, Adapter, si fuese conveniente), etc. No deben perderse de vista los conceptos de Abstracción de datos y Ocultación de la implementación, Polimorfismo o Herencia, que permiten producir código más reutilizable y más fácil de mantener y actualizar.
- **Todo el diseño de la interfaz gráfica, deberá realizarse utilizando NetBeans.**
- En todas las ventanas deberán visualizarse el **anagrama de la empresa** o de la aplicación en el margen superior izquierdo y la fecha en el derecho.
- Para poder comprobar el buen funcionamiento de la aplicación, se deberá poder arrancar con un conjunto amplio y variado de datos de prueba en la base de datos, almacenados de sesiones anteriores. Dicho conjunto deberá incluir los tipos necesarios para probar el mayor número posible de casos y opciones disponibles. Se trata de incluir todos aquellos datos de prueba que se considere oportuno para que permitan resaltar la calidad de la aplicación desarrollada.
- El sistema debe ser robusto frente a fallos y no "abortar" por circunstancias imprevistas.
- La aplicación estará **escrita en lenguaje Java**, compatible con Java2. No deberán utilizarse clases o métodos obsoletos ("deprecated").
- Los programas fuente deberán ser fáciles de mantener por cualquier programador experto en Java. Deberán estar bien estructurados y contendrán los comentarios necesarios para un fácil seguimiento del código.

Documentación a entregar junto al proyecto

La documentación, que se entregará en un documento de texto hecho con un procesador de textos, junto al proyecto en sí propuesto por el equipo de diseño, deberá incluir:

- **Portada** En ella se incluirá:
 "Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Informáticas - Modalidad a distancia".
 IES Aguadulce. Curso 2005/06
 Práctica: "<Título del proyecto elegido>."
 Autor(es):
 DNI - Apellidos, Nombre (repetido para cada uno de los alumnos, si se hace entre varios)
- **Construcción del sistema.** Este apartado incluye la documentación generada automáticamente

por Java con *JavaDoc* de cada una de las clases del sistema. Esta documentación no es necesaria incluirla en papel, tan sólo en algún directorio específico en el disco. Para generarla utilizar la herramienta "javadoc" disponible en NetBeans.

- **Manual de usuario.** En este apartado se explicará cómo interactúa un usuario con la aplicación. Es conveniente capturar las pantallas de la aplicación (se hace con las teclas Alt + ImprPant y luego se pegan en el texto) y hacer un seguimiento de las distintas opciones.
- **Manual de administración.** En este apartado se explicará cómo se puede implantar el sistema, es decir cómo instalarlo y mantenerlo, en el caso de que sea necesario tener en cuenta alguna consideración especial. Se debe indicar cuál es la clase principal que debemos ejecutar. También se debe indicar de qué ficheros habría que hacer copias de seguridad para poder recuperar los datos que maneja la aplicación en caso de desastre.
- **Casos de prueba.** En este apartado se explicarán las distintas pruebas a las que se ha sometido a la aplicación así como la información que tiene almacenado el sistema actualmente.
- **La carpeta del proyecto.** Acompañando a la documentación del proyecto, se incluirá la carpeta completa del proyecto en NetBeans, que llevará en todas las clases que formen parte del mismo los nombres de todos los autores y que contendrá:
 - Todos los archivos con **los programas fuente** (.java) de la aplicación así como el proyecto en NetBeans (guardando la carpeta del proyecto es suficiente).
 - La carpeta con la Base de Datos MySQL y los archivos necesarios para arrancar el sistema simulando un estado no inicial de funcionamiento (con datos de la BD dados de alta anteriormente), que permita comprobar que se cumplen las especificaciones formuladas.

Consideraciones

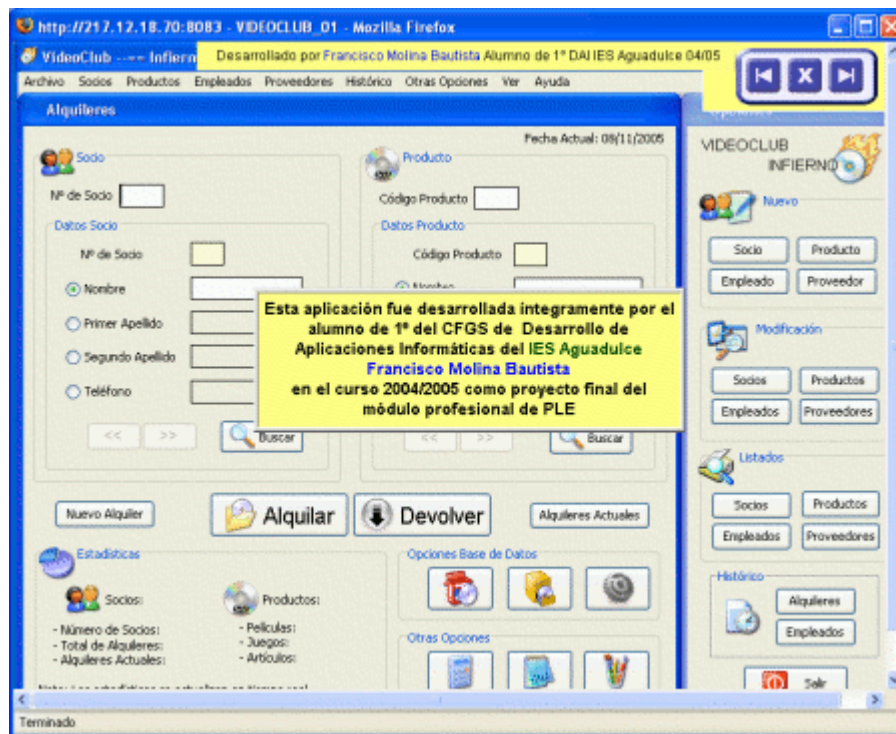
Los equipos de diseño estarán formados por una o dos personas (excepcionalmente 3). En el caso de dos o más integrantes, ambos deberán conocer todos los detalles del diseño y el funcionamiento de los programas, y especialmente de la parte que se hayan encargado de desarrollar. **Para ello, indicarán claramente y de forma destacada, los módulos que cada uno ha desarrollado, junto a la documentación entregada.**

Con el objetivo de verificar la autoría de la práctica, el tutor podrá solicitar a los diseñadores la implementación de código no recogido en el proyecto o la modificación de algún aspecto de la aplicación. Los alumnos que no sepan introducir adecuadamente las modificaciones solicitadas serán evaluados con SUSPENSO, independientemente del contenido y calidad de la práctica entregada. De la misma manera, el profesor podrá ponerse en contacto de forma individual con cualquiera de los diseñadores de la práctica, bien telefónicamente o mediante chat, para que éste le conteste a las preguntas que le formule sobre la aplicación en cuestión.

Sugerencia: Buscar en Internet aplicaciones similares que ofrezcan los mismos o parecidos servicios a los que quieres introducir en tu aplicación, para ver cómo están diseñadas y obtener ideas acerca del funcionamiento de las aplicaciones y de las interfaces de usuario.

Ejemplos guía

Todas esas especificaciones generales del apartado anterior pueden parecerle poco precisas. La idea es deliberadamente dejar el ejercicio bastante abierto y permitirte construir tu aplicación de una forma flexible al mismo tiempo que demuestras la adecuada adquisición de los conocimientos y destrezas que como programador debes poseer al terminar este módulo.



- Es posible que en la realización de tu aplicación necesites establecer algún otro requisito, que a tu juicio sea necesario. En tal caso, explícalo adecuadamente a tu tutor en la documentación que acompañe a tu aplicación, para que lo tenga en cuenta en la evaluación.
- Ten presente que cualquier suposición que hagas deberá estar justificada, y suponer algún tipo de mejora sobre lo que inicialmente se había considerado en el enunciado.

Pero para que te hagas una idea más concreta de lo que se te está pidiendo que desarrolles, te volvemos a mostrar aquí uno de los trabajos que han hecho en cursos anteriores alumnos de 1º de DAI de este módulo en nuestro centro. Seguramente te dará algunas ideas, y te aclarará bastante el trabajo, teniendo en cuenta que lo que debes conseguir es una aplicación con una funcionalidad parecida. Ten en cuenta que de lo que se trata es de que lo ejecutes y compruebes su aspecto, sus posibilidades y su funcionamiento, más que de mirar el código, que no te va a resultar del todo útil, debido a que **está desarrollado con JBuilder, que es un IDE comercial que incorpora clases propias no incluidas en el estándar de Java. Por tanto, no vas a poder usar exactamente el mismo código en NetBeans, ya que no las va a reconocer.**

Lo que te enviamos es un recopilatorio de todas las clases del proyecto ya compiladas y con una configuración de ejecución en un fichero JAR (**videoclub.jar**). Para que funcione necesitas los siguientes requisitos:

- Tener instalada correctamente una versión del jdk.
- Tener instalado MySQL, con el servidor de Bases de Datos funcionando.
- Tener disponible la base de datos **videoclub**. Te proporcionamos un **script**, es decir, un fichero de texto que contiene las sentencias sql que debes ejecutar para crear la base de datos. Ese fichero se llama **videoclub.sql**. También te proporcionamos el fichero de script **BKvideoclub.sql**, que puedes usarlo para generar un conjunto de datos de prueba en tu base de datos.

A continuación tienes todos los enlaces para descargarte y probar todo el ejemplo.

[Descarga el archivo Videoclub.jar](#)

[Descarga el archivo videoclub.sql](#)

[Descarga el archivo BKvideoclub.sql](#)

Además de la configuración de ejecución, te proporcionamos también todo el código fuente del proyecto, pero es importante que tengas en cuenta que este ejemplo fue desarrollado con un IDE distinto (JBuilder). Insistimos en que este entorno tiene la característica de que utiliza muchas clases propias para algunos elementos de las interfaces gráficas, sobre todo en lo relativo a componentes para trabajar con Bases de Datos. Esas clases propias no forman parte del estándar de Java, y por tanto no vas a poder ejecutarlo con NetBeans, ya que éste entorno no las va a reconocer. En el recopilatorio ejecutable van las mismas clases, pero ya compiladas a Bytecodes, que sí son portables.

ZONA DE DESCARGA:

Existe una versión gratuita, pero limitada en cuanto a funcionalidad, del entorno JBuilder, llamada JBuilder Foundation, que puedes descargar de la página de Borland e instalar para probar a ejecutar el ejemplo. Tiene la ventaja de que no está limitada en el tiempo, y que no necesita registro. Las otras versiones son de evaluación, por 30 días. Una vez transcurridos, no funcionará la aplicación.

Nota: Los proyectos de JBuilder se abren seleccionando el fichero .jpx del proyecto, para cargarlo en JBuilder.

[Descargas de distintas versiones de evaluación por 30 días de JBuilder, y de JBuilder Foundation](#) [\[Versión en caché\]](#)

A continuación tienes el enlace desde el que te podrás descargar la carpeta completa del proyecto de ejemplo del videoclub infierno.

[Descarga el proyecto "Videoclub Infierno"](#)



También te proporcionamos la carpeta completa del proyecto de la aplicación de la inmobiliaria InmoSoft, que también te proporcionábamos como demo de ejemplo en la unidad 4. Recuerda una vez más que de lo que se trata es de que la ejecutes y veas la funcionalidad que tiene disponible, más que de que reutilices el código, que incluye muchas clases que no forman parte del estándar de Java, y que por tanto no vas a poder usar en tu proyecto, que debe estar hecho con el entorno NetBeans. La carpeta del proyecto incluye también los scripts sql para creación de la base de datos MySQL que es necesaria para que la aplicación funcione.

[Descarga el proyecto "InmoSoft"](#)

Sugerencias de posibles proyectos a realizar

Ya hemos comentado en los apartados anteriores que lo ideal, **y lo que esperamos**, es que tú mismo defines el proyecto que quieres realizar en base a los requisitos mínimos que hemos marcado para cumplir en todos los proyectos.

Pero por si alguien no llega a tener claro qué proyecto podría realizar, **vamos a sugerirte algunos posibles proyectos**, para que al igual que Víctor en el Caso práctico del comienzo de la unidad, elijas el que más te motive, el que más atractivo te resulte, o el que vaya más en la línea de lo que piensas que te podría resultar útil. Puedes elegir cualquiera de ellos.

- En principio están pensados para hacerlos de forma individual o en grupos de dos personas, pero como están muy abiertos, la posibilidad de extenderlos y completarlos puede hacer que el tamaño aumente considerablemente, de forma que exceda la disponibilidad de tiempo real que tenéis en este módulo.
- Si realmente estás interesado en ampliar tu proyecto, y conseguir que quede terminado, completo, y con la suficiente calidad como para considerarse una aplicación "profesional", puedes ponerte de acuerdo con algún compañero, y realizar el trabajo en un grupo de dos personas (excepcionalmente de tres personas).
- También puedes abordarlos de forma individual, y hacer una versión más limitada. En cualquier caso, en la nota del proyecto se tendrá en cuenta tanto la dificultad y complejidad de la aplicación entregada como el número de personas que han intervenido en su desarrollo.

Recuerda: **previamente deberás contar con el visto bueno del tutor para hacer el trabajo en grupo, y tener en cuenta que el nivel de exigencia a la hora de realizar la evaluación del proyecto será necesariamente mayor, proporcional a la dificultad, el tamaño final y complejidad del proyecto, y en relación con el número de desarrolladores que hayan intervenido en el mismo.**

Una vez que hayas dejado volar tu imaginación un rato, valorando el proyecto que vas a realizar, el siguiente paso, es ponerte en contacto con tu tutor e indicarle qué proyecto has elegido, tanto si se trata de uno de los que te sugerimos, como si se trata del tuyo propio. En este último caso, deberás detallar a tu tutor las funcionalidades que va a incluir el mismo, para que te de el visto bueno.

Posibles proyectos

A continuación te proporcionamos los títulos de posibles proyectos que te proponemos junto al enlace que te proporcionará un enunciado de cada uno de ellos, en el que se trazan las líneas generales a incluir en el proyecto.

Gestión de un Videoclub.

Este proyecto es de una complejidad media, por lo que en principio puede ser desarrollado por un máximo de dos alumnos de forma conjunta.

El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Gestión de exámenes de una autoescuela.

Este proyecto es de una complejidad baja, por lo que en principio puede ser acometido por un único alumno. El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Gestión de reservas de habitaciones en un hotel.

Este proyecto es de una complejidad media, por lo que en principio puede ser desarrollado por dos

alumnos como máximo. El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Gestión de una Biblioteca.

Este proyecto es de una complejidad media, por lo que en principio puede ser desarrollado por dos alumnos como máximo. El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Gestión de reservas de entradas de un teatro.

Este proyecto es de una complejidad media, por lo que en principio puede ser desarrollado por dos alumnos como máximo. El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Gestión de ofertas-demandas de una inmobiliaria

Este proyecto tiene complejidad suficiente como para poder ser realizado en grupo.

El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Gestión de una tienda de ordenadores

Este proyecto es de una complejidad media, por lo que en principio puede ser acometido por un único alumno. El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Gestión de bolsa de trabajo de alumnos titulados.

Este proyecto es de una complejidad media, por lo que en principio puede ser acometido por un único alumno. El enlace siguiente te permite descargar el enunciado.

[Descarga el enunciado](#)

Otras sugerencias (I)

Seguramente has visto que lo que se pide en todos los proyectos anteriores viene a ser básicamente lo mismo:

- Administración de los datos de una empresa,
- mediante una aplicación con interfaz gráfica
- que haga inserciones, consultas, modificaciones, actualizaciones y borrados en una base de datos,
- y que use esos datos para hacer alguna operación o cálculo relacionado con su negocio.

Por eso, hemos **pensado indicarte, a modo de sugerencias, algunas ideas de otros proyectos que podrían encajar en el esquema del proyecto, para que los desarrolles tú, si te resultan interesantes.**

Gestión de una peluquería.

A primera vista una peluquería no parece tener grandes necesidades informáticas, pero es posible pensar en el control de existencias de productos usados en la misma, en el control de las ventas de determinados productos de aseo y belleza, en el control de los turnos de trabajo de los distintos trabajadores, si es una peluquería grande, y en la reserva de citas, si la peluquería tiene tanto éxito como para necesitarlo (supongamos que sí, que es poco más o menos como la de Llongueras)

Gestión de un centro educativo.

Imagina por ejemplo, que quieres hacer una aplicación para gestionar las matrículas de los alumnos del ciclo de DAI, que sabes los módulos que tiene, con toda la información de los alumnos, de los padres, de los profesores, y de las notas...

Cuestiones a tener en cuenta: no se puede pasar a segundo con el 25% de horas supensas en los módulos de primero (hablamos de modalidad presencial), no se puede repetir más que un año cada módulo, y de cada módulo debe ser posible anotar para la evaluación la nota de tres evaluaciones, más la nota final de la convocatoria ordinaria, y la nota de la convocatoria extraordinaria para los que la necesiten. No podrás evaluar a un alumno más que en los módulos que está matriculado, y no podrás evaluarlo de una evaluación o de una convocatoria si no tiene evaluadas las anteriores. Además, sólo se admite evaluar la extraordinaria si la ordinaria estaba suspensa...

Se puede considerar el control de faltas de asistencia, justificadas y sin justificar (con el 20 % de faltas sin justificar, se pierde el derecho a las evaluaciones intermedias, pero no a las convocatorias finales. Lo mismo ocurre con el 30 % de faltas, aunque sean justificadas, por imposibilidad de aplicar evaluación continua a alguien que falta tanto, sea por la causa que sea). Bueno, se pueden ir añadiendo cosas, que seguro que vuestro conocimiento como alumnos os permite añadir más cosas.

Otras sugerencias (II)

Gestión de las tareas y los turnos de trabajo en una gran empresa.

En una gran empresa dedicada a la fabricación de productos químicos, tienen turnos de trabajo las 24 horas del día todos los días del año. La plantilla tiene un número alto de trabajadores, y para no perjudicar a nadie con los peores turnos, se establecen turnos rotatorios de 8 horas de mañana, tarde y noche.

Es necesario **gestionar** esos turnos, de forma que:

- todos los trabajadores completen su jornada laboral de 40 horas semanales,
- que se tengan en cuenta las vacaciones de cada cual,
- intentando en la medida de lo posible, cumplir las preferencias de los trabajadores, de forma que todos los puestos de trabajo estén cubiertos las 24 horas del día de todos los días del año,
- con posibilidad de gestionar cambios de turnos de mutuo acuerdo entre los trabajadores,
- pero de forma que ningún trabajador pueda hacer dos turnos consecutivos, por cuestiones de seguridad, al ser un trabajo pesado que requiere atención, y con cierto grado de peligro asociado.
- Deben tenerse en cuenta mecanismos para cubrir las bajas por enfermedad,
- y para gestionar un sistema de horas extras, de forma que se puedan atender imprevistos, y que se cumplan los requisitos de seguridad que obligan a que no pueda haber turnos de más de 8 horas, ni dos turnos consecutivos...
- Se puede gestionar el pago de la nómina de los trabajadores de acuerdo al puesto que desempeñan, a los turnos que cubren (la noche cobra complemento de nocturnidad) y las horas extras realizadas, más algún complemento que se pueda considerar en función de antigüedad, categoría profesional, número de hijos, etc....

Gestión de almacén

Se trata de **controlar las existencias** en un almacén, llevando el control del stock, de los productos, de los clientes y de los proveedores.

- Habrá compras de productos a proveedores,
- ventas de productos a clientes,

- facturación y contabilidad de compras y ventas, etc.

Interesa saber el proveedor que nos suministra cada producto, y los productos que compra cada cliente. También controlar que las existencias de cada producto no bajen de un nivel crítico, que si lo superan se genere un pedido automáticamente para el proveedor adecuado (el que en ese momento suministre el producto a mejor precio).

Gestión de una ferretería.

Realmente se trata de un problema bastante parecido al de gestión de Stock de un almacén, ya que puede verse la ferretería como un caso particular de almacén, que maneja un tipo particular de productos. Las diferencias principales pueden venir por el **tipo de clientes** que se tienen. En un almacén probablemente la mayoría de los clientes son mayoristas, mientras que la ferretería trabaja más con la venta minorista y detallista. Eso conlleva una menor necesidad de control de datos de clientes, ya que los clientes particulares no suelen solicitar facturas. Como digo, puede verse como una versión ligeramente simplificada del ejemplo de control de Almacén.

Otras sugerencias (III)

Gestión de una guardería.

Se trata de tener registrados todos los datos de los niños que hay en la guardería que puedan ser de interés (nombre, edad, enfermedades, alergias, consideraciones sobre necesidad de cuidados especiales, datos de los padres, teléfono para comunicar emergencias, personas que normalmente o esporádicamente pasarán a recogerlo, etc.)

Se puede llevar un diario con las actividades realizadas diariamente por el niño, así como de cualquier circunstancia destacable, que se le puede entregar diariamente a los padres al recogerlo, además de permanecer registrado en la aplicación. Se puede tener la información sobre los cuidadores de la guardería, relacionando cada niño con su cuidador, establecer una serie de horarios disponibles, y registrar el horario de cada niño, en función del cual se facturará a final de mes. También se puede llevar el control de gastos de la guardería, incluyendo material utilizado y gastos de personal. El resto, para que lo desarrolléis vosotros.

Gestión de un concesionario.

Vuelve a ser la típica aplicación en la que se debe registrar la información de los coches en stock para vender, de los clientes, y de los trabajadores del concesionario. Se pueden considerar la gestión tanto de:

- las **ventas** de vehículos nuevos,
- vehículos de ocasión,
- y gestión de las reparaciones del taller.

Cada cosa requerirá mantener diferente tipo de información. Para las ventas de **coches nuevos**, el precio del coche, los datos del cliente, el vendedor que hizo la venta, y saber si la matriculación, y tramitación de la documentación necesaria la hace el propio cliente o la hace el propio concesionario. Para los **coches de ocasión**, además de los datos anteriores, interesará conocer la fecha de matriculación, si tiene las ITV's en regla, si tiene pendiente el abono de algún tipo de sanción o impuesto por parte del anterior dueño, etc.

Para la **gestión del taller**, habrá que considerar todo lo relativo a las averías y revisiones periódicas, considerando los tiempos medios de reparación, precio de la mano de obra, precio de las piezas usadas, el cobro o no de gastos de depósito y custodia en caso de que el cliente tarde más de lo debido en retirar el coche del taller una vez reparado, etc.

Gestión de un club deportivo.

Se trata de tener registrados los datos de todas las instalaciones del club, que pueden ser pistas

deportivas de tenis, fútbol, fútbol sala, padel, baloncesto, voleibol, piscina, gimnasio, salón de juegos (billar, pimpón), etc. Cada tipo de instalación deportiva necesitará tener registrada información distinta sobre su capacidad, restricciones de uso, etc. Estas instalaciones podrán reservarse para socios particulares del club que harán uso de ellas, y que se les facturará junto a su cuota mensual, según la tarifa vigente. También podrán ser reservadas las instalaciones para cursos de las distintas escuelas deportivas que tiene en marcha el club, de forma que habrá que relacionar para cada curso, la instalación reservada, el horario de la reserva, el monitor encargado del curso, y la lista de alumnos asistentes a ese curso.

También se podrán reservar para competiciones, en cuyo caso habrá que mantener la información relativa a las competiciones en el club (participantes, árbitros, instalaciones y horario, etc.)

Este ejemplo da mucho de sí, y se puede ampliar bastante, por lo que puede ser realizado perfectamente por un grupo de tres desarrolladores.

Otras sugerencias (IV)

Gestión de cursos, profesorado y alumnado de una academia.

Es parecido al anterior, en el sentido de que también hay que gestionar unos cursos, que imparten unos profesores, y a los que se apuntan unos alumnos, previo pago de una matrícula y de una cuota mensual que habrá que facturarles. Un profesor podrá impartir varios cursos, pero cada curso se impartirá por un solo profesor. Un alumno podrá matricularse en más de un curso, y por supuesto un curso tiene varios alumnos. Cada curso estará asociado a un calendario y a un horario. Cada alumno, al final del curso, obtendrá una calificación, que le dará derecho a la obtención de un diploma si es positiva, o no. Se trata de gestionar toda esa información desde nuestra aplicación.

Gestión de obras de una constructora.

La constructora tendrá que gestionar los gastos de las distintas obras que tiene en construcción. Se trata de llevar una gestión de los costes de cada obra, para lo que habrá que controlar los materiales empleados, los trabajadores que intervienen, y las herramientas y maquinaria empleadas.

La constructora dispone de una plantilla de trabajadores, que incluye diferentes tipos de profesionales (peones albañiles, encofradores, electricistas, fontaneros, aparejadores, encargados de obra, arquitectos, etc), cada uno de los cuales tiene un sueldo diferente. Los distintos profesionales van pasando por las distintas construcciones en marcha, según la fase en la que se encuentren, por lo que los jornales de cada día de cada empleado deben ser imputados a la obra en la que trabajan ese día. Igual ocurre con la maquinaria y las herramientas, que se irán rotando por las distintas obras de la empresa según su fase de construcción. Esas herramientas también tienen un coste diario de alquiler, o un coste de amortización, si es que son de su propiedad, por lo que deberá controlarse la obra en la que está cada maquinaria y cada herramienta diariamente, para imputar ese coste al coste general de la obra. Se puede considerar también que las herramientas y maquinaria pueden estar temporalmente sin uso, por lo que habrá que mantenerlas almacenadas, y habrá un coste medio de almacenaje, que habrá que prorratear entre todas las obras que estén en marcha durante el periodo que estén almacenados. Por tanto, debemos saber dónde se encuentra (obra o almacén) cada día cualquier herramienta, o maquinaria. También dónde se encuentra cada día cada trabajador. Transitoriamente pueden estar en paro, pero eso se supone que no tiene ningún coste adicional para la empresa constructora.

La aplicación, debe ofrecer en cada momento la información sobre el coste acumulado hasta ese día por cada construcción, o del coste total si ya está terminada, y cualquier otra información, que pueda ser de utilidad.

Gestión de reservas de eventos en un hotel.

En un hotel hay varios salones destinados a celebración de distintos tipos de eventos, como congresos, reuniones de empresas, bodas, bautizos, comuniones, etc. Para cada evento, los contratantes solicitan determinados servicios, de entre los que el hotel les ofrece. Esos servicios incluyen la elección de un menú personalizado, mediante la elección de los aperitivos, los distintos platos, vinos, copas, barra libre,

etc. de entre los disponibles, que ofrecerá la aplicación. También habrá otros servicios que se facturarán, como amenizar con música, habilitar pista de baile, contratación, animación para los niños, etc.

La aplicación gestionará las reservas de esos eventos, (consultas, altas, bajas y modificaciones) y ofrecerá la facturación según las opciones contratadas.

Guía de evaluación del proyecto

¿Cómo vamos a evaluar el proyecto?

Son muchas las cosas que podemos valorar en un proyecto, pero seguramente te resultará de utilidad saber los criterios que se van a seguir de forma general en la evaluación.

Lo primero que tenemos que decir es que la valoración se hará de una forma global. La calidad del proyecto y de la aplicación se tiene que valorar en su conjunto. No obstante, como este proyecto no tiene una finalidad comercial, sino que su propósito es valorar la consecución de los objetivos de este módulo profesional, los puntos en los que nos vamos a fijar pueden ser ligeramente distintos de los que te exigiría el cliente al que le intentas vender una aplicación.

A continuación te vamos a dar las indicaciones que nosotros mismos vamos a seguir a la hora de evaluar y poner nota a tu proyecto. Se trata de indicarte los puntos en los que nos vamos a fijar y la importancia relativa dentro del total que le vamos a conceder a cada uno.

Presentación y documentación

Procura que tu proyecto sea claro, que en la presentación cumplas con todos los requisitos que se te han solicitado, y que esté bien documentado.

Ya te hemos indicado anteriormente la documentación que te solicitamos. Ten en cuenta que el tiempo del que dispones es escaso, y hacer una completa documentación es algo que lleva mucho tiempo. Pero se te pide que el código esté en la medida de lo posible autocomentado, mediante comentarios javadoc y que generes la documentación a partir de esos comentarios. También te pedimos que cuides los aspectos de indentación del código y que respetes los convenios de nomenclatura establecidos para los identificadores.

Especialmente importante es la confección de un "manual de usuario". Ten en cuenta que no debe ser exhaustivo (por abreviar, ya que no hay mucho tiempo) pero sí debe servir para que el tutor al corregir pueda consultar las opciones disponibles, y como realizar cada una de ellas, sin tener que perder mucho tiempo "trasteando" la aplicación. También debe incluir cualquier consideración que estimes oportuna sobre la forma de hacer las cosas la aplicación.

La valoración de este apartado en la nota global del proyecto será de un 15%

Defensa del proyecto

Aquí no es posible que cada uno de vosotros haga una defensa presencial de su proyecto ante su tutor. Pero sí deberíais elaborar un breve documento en el que detalléis los mejores aspectos de vuestra aplicación, y aquellos detalles que pueden haberos llevado mucho tiempo de desarrollo pero cuyo resultado no siempre es llamativo.

También podéis dar una lista de posibles **mejoras** para vuestra aplicación, que han tenido que quedar aparcadas por falta de tiempo. Con esto pretendemos saber si lo que no has incluido es por desconocimiento o sencillamente por falta de tiempo.

La valoración de este apartado en la nota global del proyecto será de un 5%

Aplicación desarrollada

La parte más importante de la nota, como no podía ser de otra manera, se la llevará la aplicación propiamente dicha, que se valorará como el **80% restante de la nota**.

No obstante, hay muchos apartados que se valorarán dentro de la aplicación. Te los enumeramos a continuación, con una indicación de su valoración global dentro de la nota del proyecto. Debes tener en cuenta que la mayoría de esos apartados están íntimamente relacionados entre sí, por lo que no es fácil separar a veces lo que es un fallo de la funcionalidad de lo que es un fallo del diseño del interfaz gráfico, o lo que es un fallo del acceso a los datos de la base de datos.

Funcionalidad adecuada y ejecución correcta (sin errores o al menos sin errores graves.)

Se valorará que la aplicación resuelva correctamente todos los aspectos importantes del problema. Por ejemplo, no sería admisible una aplicación que no permitiera dar de alta a nuevos clientes de un negocio, o que no permitiera corregir un fallo al introducir los datos de alguno de ellos. También sería considerado un error grave que la aplicación abortara para determinados datos de prueba, o bajo determinadas circunstancias, aunque funcionara correctamente para otros datos.

En definitiva, en este apartado se trata de ver que la **aplicación hace bien todo lo que tiene que hacer**.

La valoración de este apartado en la nota global del proyecto será de un 35%

Diseño de un interfaz gráfico completo, útil, amigable y adecuado al problema.

En este apartado valoraremos que se hayan elegido los componentes Swing más adecuados para cada funcionalidad de la aplicación. Por ejemplo, no será lo mismo que un listado de clientes se muestre en un área de texto como un único String, a que se muestre en un JTable, con cada cliente en una fila de la tabla, y en la cabecera de la tabla indicado el nombre del campo o atributo de ese cliente al que corresponde cada cliente.

Por ejemplo, no es lo mismo que al hacer una búsqueda por puesto de trabajo yo tenga que incluir a mano el nombre del puesto a buscar a que pueda introducirlo seleccionándolo de una lista o de una lista desplegable.

También habrá que valorar que la interacción con las ventanas y componentes de la aplicación se pueda hacer de forma cómoda, para que realmente suponga una ventaja el uso de la aplicación para el problema en cuestión. Y aunque en menor medida, también se valorará que el aspecto de la aplicación sea agradable, y que el diseño de las ventanas sea intuitivo y por tanto la aplicación fácil de usar.

La valoración de este apartado en la nota global del proyecto será de un 20%

Incorporación del mayor número de elementos de entre los estudiados, con un uso adecuado de ellos.

Puesto que se trata de evaluar el aprendizaje que has hecho de lo aprendido en el curso, es importante que demuestres que sabes usar los distintos componentes Swing que se han visto. Evidentemente, en determinados casos, es posible resolver un determinado problema de forma igualmente satisfactoria usando diferentes tipos de componentes Swing. Se trata de que si tienes ocasión, incluyas todos los tipos, en vez de limitarte a usar siempre el mismo. De la misma manera, la aplicación puede ser correcta sin incluir algún tipo concreto de funcionalidad, que no es muy importante, y que no es imprescindible. Pero si incluir esa nueva funcionalidad te permite usar algún componente que no habías usado hasta el momento, sería conveniente incluirla.

La valoración de este apartado en la nota global del proyecto será de un 5%

Incorporación de acceso a Bases de Datos (preferiblemente a MySQL con JDBC) y diseño adecuado de consultas, borrados y actualizaciones.

En definitiva se trata de ver que la aplicación conecta con la base de datos, y efectivamente nos permite consultar, actualizar y borrar datos, de forma que todas las operaciones que hagamos en nuestra aplicación tengan su reflejo en la Base de Datos, y viceversa.

La valoración de este apartado en la nota global del proyecto será de un 15%.

Un paso más: ampliación y aportación de elementos y características investigadas autónomamente por el alumno, que no se han estudiado en el módulo profesional, o que se han explicado de forma somera.

Ésta siempre va a ser una de las principales tareas de tu vida profesional como programador, ya que siempre vas a tener que investigar nuevas herramientas, nuevos lenguajes, nuevas funciones, nuevos componentes, nuevas formas de hacer las cosas, en definitiva. Por tanto se trata de que demuestres tu capacidad para investigar por tu cuenta, para aprender algo más allá de lo que te han enseñado, ya que en el trabajo, seguramente poco de todo lo que has aprendido en este módulo te resultará útil dentro de 15 o 20 años.

La valoración de este apartado en la nota global del proyecto será de un 5%

Últimas palabras

Aquí termina este módulo profesional.

Ha sido un placer compartir contigo el camino que nos ha traído hasta aquí.

Esperamos que te haya resultado enriquecedor. Sobre todo que lo hayas disfrutado, y encontrado útil, con independencia de la nota que finalmente obtengas, y que siempre hayas encontrado en nosotros la ayuda necesaria.

¡Mucha suerte con la realización de tu proyecto!