



Sistema de Gestión de Contenidos

Contenidos de la propuesta de proyecto

Índice

1. Presentación del proyecto

- Enunciado de la práctica
- La Formación Profesional
- Desarrollo de Aplicaciones Web
- Anteproyecto presentado
- Los Gestores de Contenidos y las web dinámicas

2. Realización del proyecto

- El uso de frameworks
- Modelo Vista Controlador
- Code Ignater
- Documentación y nomenclatura

Normas para la realización del módulo de proyecto del ciclo de DAW

(Base legislativa B.O.E. Real Decreto 1147/2011.- Artículo 26.)

1. Los ciclos formativos de grado superior incorporarán un módulo profesional de proyecto, que se definirá de acuerdo con las características de la actividad laboral del ámbito del ciclo formativo y con aspectos relativos al ejercicio profesional y a la gestión empresarial.
2. El módulo tendrá por objeto la integración de las diversas capacidades y conocimientos del currículo de todos los módulos del ciclo formativo. Esta integración se concretará en proyectos que contemplen las variables tecnológicas y organizativas relacionadas con el título. Para ello, el módulo profesional de proyecto se realizará durante el último período del ciclo formativo y su evaluación quedará condicionada a la evaluación positiva del resto de los módulos profesionales del ciclo formativo, incluido el de formación en centros de trabajo.
3. Este módulo se organizará sobre la base de la tutoría individual y colectiva y su atribución docente correrá a cargo del profesorado que imparta docencia en el ciclo formativo, preferentemente en módulos profesionales asociados a unidades de competencia.

1.- Presentación y valoración de la propuesta

Los proyectos podrán ser propuestos por el alumnado o por el departamento.

Cuando sea planteado por el alumno, se presentará una propuesta sobre el contenido del proyecto que se propone realizar, por lo menos, un mes antes de la fecha de realización de la evaluación previa a la incorporación al primer período de realización del módulo de FCT.

El equipo docente del grupo presidido por el tutor, valorará la propuesta del proyecto presentada por cada alumno, teniendo en cuenta, al menos, su adecuación a los contenidos abordados en el ciclo formativo y la posibilidad de realización efectiva del proyecto en el plazo previsto.

2.- Aceptación de la propuesta o asignación de proyectos.

El equipo docente del grupo procederá a la aceptación de los proyectos propuestos por el alumnado o a la asignación de proyectos, haciéndolo constar en un acta que se publicará en el tablón de anuncios del centro docente.

El alumnado cuya propuesta de proyecto no fuese aceptada por el equipo docente dispondrá de un plazo de quince días hábiles, después de notificadas las causas de manera individualizada al alumnado, para introducir las modificaciones oportunas o proceder a la presentación de una nueva propuesta.

Si no se presentan modificaciones o una nueva propuesta que fuese aceptada, se considerará que el alumno renuncia a la convocatoria del módulo.

3.- Registro de la propuesta

Una vez repartidos los proyectos, se registrará la propuesta en secretaría al menos dos meses antes de la fecha que el centro establezca para su exposición o defensa.

Una vez registrada la propuesta el alumno o alumna podrá solicitar la renuncia a la convocatoria, según se establece en la normativa vigente sobre evaluación.

4.- Entrega, exposición y defensa

Aceptada la propuesta del proyecto o la asignación del mismo, el tutor individual fijará las fechas en las que el alumnado deberá entregar, exponer y defender el proyecto.

Las fechas serán publicadas en el tablón de anuncios del centro.

Evaluación y calificación

Tribunal de evaluación

Tutor colectivo, tutor individual y jefe de departamento.

Si ambas tutorías son ejercidas por la misma persona, la dirección del centro designará un profesor del equipo docente del grupo.

Con el fin de incorporar competencias adquiridas durante la realización de las FCT's, se evaluará una vez esta haya sido cursada.

Calificación del proyecto

Contenidos: dificultad, grado de resolución de la propuesta, originalidad, actualidad, alternativas presentadas y resultados obtenidos.	50%
Aspectos formales: presentación, estructura, organización, redacción del proyecto	20%
Exposición y defensa: calidad de la exposición oral y de las respuestas a las preguntas planteadas por los miembros del tribunal.	30%

Cada miembro emitirá una calificación sobre cada apartado, obteniéndose la media de ellas en cada caso.

La calificación final será la media ponderada de los diversos apartados, sin decimales.

Si el proyecto no obtuviera una calificación positiva en su primer periodo de realización el tribunal elaborará un informe en el que consten los defectos que deban ser subsanados.

En cada uno de los periodos de realización del módulo de Proyecto, el alumnado dispondrá de convocatoria ordinaria y extraordinaria.

La Formación Profesional

Con el objetivo de hacer accesible este documento a todos los públicos, independientemente de sus conocimientos en los aspectos ajenos a la simple programación, se realizará una breve introducción sobre los principales aspectos de la Formación Profesional y posteriormente, sobre los del Desarrollo de Aplicaciones Web. Para ello se han incluido varios artículos del BOE que hacen referencia a nuestros objetos de interés.

13118 - Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo.

Artículo 4. La ordenación de la formación profesional.

Las enseñanzas de la formación profesional del sistema educativo se ordenan en:

- a) Los módulos profesionales específicos de los programas de cualificación profesional inicial.
- b) Los ciclos formativos de grado medio.
- c) Los ciclos formativos de grado superior.
- d) Los cursos de especialización.

Artículo 5. Organización de las enseñanzas.

1. Las enseñanzas de formación profesional del sistema educativo deben responder a un perfil profesional y se organizan en módulos profesionales de duración variable.
2. Las enseñanzas de formación profesional se organizarán de manera que permitan la conciliación del aprendizaje de las personas con otras actividades y responsabilidades.
3. Las enseñanzas de formación profesional se adaptarán al alumnado con necesidad específica de apoyo educativo para que se garantice su acceso, permanencia y progresión en estas enseñanzas.

Artículo 6. Los módulos profesionales.

1. Los módulos profesionales estarán constituidos por áreas de conocimiento teóricoprácticas, en función de las competencias profesionales, sociales y personales que se pretendan alcanzar. Estos módulos profesionales, según su naturaleza, estarán asociados o no a unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales.
2. Con el fin de promover la formación a lo largo de la vida, las Administraciones educativas podrán organizar la impartición de los módulos profesionales en unidades formativas de menor duración. Estas unidades podrán ser certificables. La certificación tendrá validez en el ámbito de la correspondiente Administración educativa. La superación de todas las unidades formativas que constituyen un módulo profesional dará derecho a la certificación del mismo, que es la unidad mínima de certificación con valor en todo el territorio nacional.

Artículo 7. El perfil profesional.

Los elementos que definen el perfil profesional de cada enseñanza son los siguientes:

- a) La competencia general. Describe las funciones profesionales más significativas del perfil profesional. Tomará como referente el conjunto de cualificaciones profesionales y las unidades de competencia incluidas. En los cursos de especialización, la competencia general podrá estar referida al Catálogo Nacional de Cualificaciones Profesionales.

- b) Las competencias profesionales, personales y sociales. Describen el conjunto de conocimientos, destrezas y competencia, entendida ésta en términos de autonomía y responsabilidad, que permiten responder a los requerimientos del sector productivo, aumentar la empleabilidad y favorecer la cohesión social.
- c) Las cualificaciones profesionales y, en su caso, las unidades de competencia cuando se refieran al Catálogo Nacional de Cualificaciones Profesionales.

Artículo 8. El currículo.

1. Corresponde al Gobierno, mediante real decreto, establecer los aspectos básicos del currículo que constituyen las enseñanzas mínimas de los ciclos formativos y de los cursos de especialización de las enseñanzas de formación profesional que, en todo caso, se ajustarán a las exigencias derivadas del Sistema Nacional de Cualificaciones y Formación Profesional.
2. Las Administraciones educativas establecerán los currículos correspondientes respetando lo dispuesto en el presente real decreto y en las normas que regulen las diferentes enseñanzas de formación profesional. En todo caso, la ampliación y desarrollo de los contenidos incluidos en los aspectos básicos del currículo, establecido por el Gobierno, se referirán a las cualificaciones y unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales incluidas en las correspondientes enseñanzas, así como a la formación no asociada a dicho catálogo, respetando el perfil profesional establecido.
3. Las Administraciones educativas tendrán en cuenta, al establecer el currículo de las enseñanzas reguladas en el presente real decreto, la realidad socioeconómica del territorio de su competencia, así como las perspectivas de desarrollo económico y social, con la finalidad de que las enseñanzas respondan en todo momento a las necesidades de cualificación de los sectores socio-productivos de su entorno, sin perjuicio alguno a la movilidad del alumnado. Para ello, se contará con la colaboración de los interlocutores sociales.
4. Los centros de formación profesional aplicarán los currículos establecidos por la Administración educativa correspondiente, de acuerdo con las características y expectativas del alumnado, con especial atención a las necesidades de aquellas personas que presenten una discapacidad. Asimismo, se tendrán en cuenta las posibilidades formativas del entorno, especialmente en el módulo profesional de formación en centros de trabajo.
5. Con el fin de facilitar al alumnado la adquisición de las competencias correspondientes, las Administraciones educativas, en el marco de sus competencias, promoverán la autonomía pedagógica organizativa y de gestión de los centros que impartan formación profesional, fomentarán el trabajo en equipo del profesorado y el desarrollo de planes de formación, investigación e innovación en su ámbito docente, así como las actuaciones que favorezcan la mejora continua de los procesos formativos.
6. La metodología didáctica de las enseñanzas de formación profesional integrará los aspectos científicos, tecnológicos y organizativos que en cada caso correspondan, con el fin de que el alumnado adquiriera una visión global de los procesos productivos propios de la actividad profesional correspondiente.

Artículo 9. Estructura de los títulos de formación profesional y de los cursos de especialización.

Las disposiciones estatales que establezcan un título de formación profesional o un curso de especialización, y determinen sus características y contenidos básicos, deberán tener, como mínimo, el siguiente contenido:

- a) Identificación del título o curso de especialización:
 - Denominación.
 - Nivel en el sistema educativo.
 - Duración.
 - Familia o familias profesionales.

- Nivel en el Marco Español de Cualificaciones y, para los ciclos formativos de grado superior, además, nivel del Marco Español de Cualificaciones para la Educación Superior y sus correspondencias con los marcos europeos.
- b) Perfil profesional. Competencia general, competencias profesionales, personales y sociales. Relación de cualificaciones profesionales y unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales incluidas.
- c) El entorno profesional, que incluye, entre otros, las ocupaciones y puestos de trabajo.
- d) La prospectiva en el sector o sectores.
- e) Enseñanzas:
 - Objetivos generales.
 - Módulos profesionales, en los términos previstos en el artículo siguiente.
- f) Los parámetros básicos de contexto formativo. Se concretarán: los espacios y los equipamientos mínimos, adecuados al número de puestos escolares, así como las titulaciones y especialidades del profesorado, y sus equivalencias a efectos de docencia.
- g) La correspondencia, en su caso, de los módulos profesionales con las unidades de competencia para su acreditación.
- h) Convalidaciones, exenciones y equivalencias.
- i) Información sobre los requisitos necesarios según la legislación vigente para el ejercicio profesional.
- j) Para los títulos de grado superior, la modalidad y materias del Bachillerato que faciliten la admisión en caso de concurrencia competitiva.

Artículo 10. Estructura de los módulos profesionales.

1. Los módulos profesionales incluirán, en su caso, las especificaciones de la formación recogidas en los correspondientes módulos formativos del Catálogo Nacional de Cualificaciones Profesionales.
2. Los módulos profesionales estarán definidos en resultados de aprendizaje, criterios de evaluación y contenidos, tomando como referencia las competencias profesionales, personales y sociales que se pretenden desarrollar a través del módulo profesional.
3. El real decreto por el que se establezca un título de formación profesional o un curso de especialización especificará para cada módulo profesional los siguientes aspectos:
 - a) Denominación y código.
 - b) Los objetivos expresados en resultados de aprendizaje.
 - c) Criterios de evaluación.
 - d) Contenidos básicos del currículo, que quedarán descritos de forma integrada en términos de procedimientos, conceptos y actitudes. Se agruparán en bloques relacionados directamente con los resultados de aprendizaje.
 - e) Orientaciones pedagógicas.
 - f) Duración mínima en horas del módulo profesional en la modalidad presencial.
 - g) Número de créditos ECTS de cada módulo profesional en los ciclos formativos de grado superior y cursos de especialización.
 - h) Requisitos del profesorado.
4. Los módulos profesionales específicos de los PCPI se ajustarán a la misma estructura, en los apartados que les afecten.
5. La adecuación puntual de módulos profesionales, incluidos en las diferentes ofertas de formación profesional del sistema educativo, a los cambios de las unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales se hará por orden ministerial. Las Administraciones educativas comunicarán a los centros docentes las adaptaciones que deban realizarse en el currículo de los módulos profesionales que resulten afectados por estas actualizaciones.

Desarrollo de Aplicaciones Web

17329 - Orden EDU/2887/2010, de 2 de noviembre, por la que se establece el currículo del ciclo formativo de Grado Superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web.

Artículo 3. Currículo.

1. El currículo para las enseñanzas de formación profesional del sistema educativo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web establecido en el Real Decreto 686/2010, de 20 de mayo, queda determinado en los términos fijados en esta Orden.
2. El perfil profesional del currículo, que viene expresado por la competencia general, las competencias profesionales, personales y sociales y las cualificaciones y las unidades de competencia del Catálogo Nacional de Cualificaciones Profesionales, es el incluido en el título de Técnico Superior en Desarrollo de Aplicaciones Web referido en el punto anterior.
3. Los objetivos generales del currículo del ciclo formativo, los objetivos de los módulos profesionales expresados en términos de resultados de aprendizaje y sus criterios de evaluación son los incluidos en el título de Técnico Superior en Desarrollo de Aplicaciones Web referido en el punto 1 de este artículo.
4. Los contenidos de los módulos profesionales que conforman el presente currículo, adaptados a la realidad socioeconómica así como a las perspectivas de desarrollo económico y social del entorno, son los establecidos en el Anexo I de esta Orden.

Artículo 4. Duración y secuenciación de los módulos profesionales.

1. La duración total de las enseñanzas correspondientes a este ciclo formativo, incluido el módulo profesional de formación en centros de trabajo, es de 2.000 horas.
2. Los módulos profesionales de este ciclo formativo, cuando se oferten en régimen presencial, se organizarán en dos cursos académicos y se ajustarán a la secuenciación y distribución horaria semanal determinadas en el Anexo II de esta Orden.
3. El primer curso académico se desarrollará íntegramente en el centro educativo. Para poder cursar el segundo curso, será necesario haber superado los módulos profesionales que supongan en su conjunto, al menos, el ochenta por ciento de las horas del primer curso y, en cualquier caso, todos los módulos profesionales soporte incluidos en el mismo, señalados como tales en el Anexo II.
4. Se garantizará el derecho de matriculación de quienes hayan superado algún módulo profesional en otra Comunidad Autónoma en los términos establecidos en el artículo 31.3 del Real Decreto 1538/2006.
5. Con carácter general, durante el tercer trimestre del segundo curso, y una vez alcanzada la evaluación positiva en todos los módulos profesionales realizados en el centro educativo, se desarrollará el módulo profesional de formación en centros de trabajo.
6. Excepcionalmente, y con el fin de facilitar la adaptación del número de personas matriculadas a la disponibilidad de puestos formativos en las empresas, aproximadamente la mitad del alumnado de segundo curso podrá desarrollar dicho módulo profesional de formación en centros de trabajo durante el segundo trimestre del segundo curso, siempre y cuando hayan superado positivamente todos los módulos profesionales del primer curso académico.
7. Sin perjuicio de lo anterior y como consecuencia de la temporalidad de ciertas actividades económicas que puede impedir que el desarrollo del módulo profesional de

formación en centros de trabajo pueda ajustarse a los supuestos anteriores, éste se podrá organizar en otros períodos coincidentes con el desarrollo de la actividad económica propia del perfil profesional del título.

8. En cualquier caso, la evaluación del módulo profesional de formación en centros de trabajo quedará condicionada a la evaluación positiva del resto de los módulos profesionales del ciclo formativo.

Artículo 5. Módulo profesional de proyecto.

1. El módulo profesional de proyecto tiene un carácter interdisciplinar e incorpora las variables tecnológicas y organizativas relacionadas con los aspectos esenciales de la competencia profesional del título de Técnico Superior en Desarrollo de Aplicaciones Web.

2. Con carácter general este módulo será impartido por el profesorado que ejerce la tutoría de formación en centros de trabajo.

3. El módulo profesional de proyecto se desarrollará durante el mismo periodo que el módulo profesional de formación en centros de trabajo. El profesorado responsable de su desarrollo deberá anticipar las actividades de enseñanza y aprendizaje que faciliten el desarrollo posterior del módulo.

4. El desarrollo y seguimiento del módulo profesional de proyecto deberá compaginar la tutoría individual y colectiva. En cualquier caso, al menos el 50% de la duración total se llevará a cabo de forma presencial, completándose con la tutoría a distancia empleando las tecnologías de la información y la comunicación.

5. La evaluación de este módulo profesional quedará condicionada a la evaluación positiva del resto de los módulos profesionales del ciclo formativo, incluido el de formación en centros de trabajo.

Anteproyecto

1. Presentación:

A continuación se adjunta el documento presentado y aprobado como anteproyecto ante el tutor. Dicho anteproyecto no contiene más información que la ya presentada, así que si ya se ha leído no será necesaria su revisión.

2. Descripción.

- Objetivos del proyecto, sector al que está dedicado.
Crear un Sistema de Gestión de Contenidos mediante el cual se puedan publicar noticias, con la posibilidad de administrar un foro y anunciar eventos mediante un calendario.

Además, el administrador podrá distribuir el menú y las secciones a su antojo, incluyendo las que crea necesarias en cualquier parte de la retícula y modificar el diseño entre una gama de plantillas.

Finalmente deberá ser posible subir imágenes a la base de datos, no con la intención de crear una galería, pero sí para que puedan ser utilizadas en noticias.

- Características básicas.- Imprescindibles para que el proyecto sea aceptado
Elementos que el Administrador podrá añadir:
 - Páginas estáticas: Son páginas cuya información no varía. En ella el administrador podrá incluir información sobre el portal web, el FAQ.
 - Página de noticias: Páginas dinámicas que mostrarán las noticias de un o más temas.
 - Categorías del foro: Los topics de un foro están organizados por categorías, separaciones que engloban un conjunto de topics de una misma temática. El administrador debe ser libre de crear esos temas.
 - Podrá administrar a los usuarios, pudiendo asignar los diferentes roles entre ellos.

Opciones disponibles para el invitado:

- Podrá acceder a las secciones públicas de la página
- El invitado podrá leer las noticias publicadas
- Si el administrador decide colocar el calendario en una página publica, el invitado podrá ver los eventos programados
- Si el administrador decide colocar el foro en una página publica, el invitado podrá ver los topics y posts del foro

Opciones disponibles para el registrado:

- Tendrá acceso a todas las opciones del invitado
- Podrá comentar noticias y eventos
- Podrá generar nuevos posts en el foro, ya sean nuevos temas o respuestas a temas ya existentes

Opciones disponibles para el redactor:

- Tendrá acceso a todas las opciones del registrado
- Podrá redactar nuevas noticias
- Podrá crear nuevos eventos

Opciones disponibles para el editor:

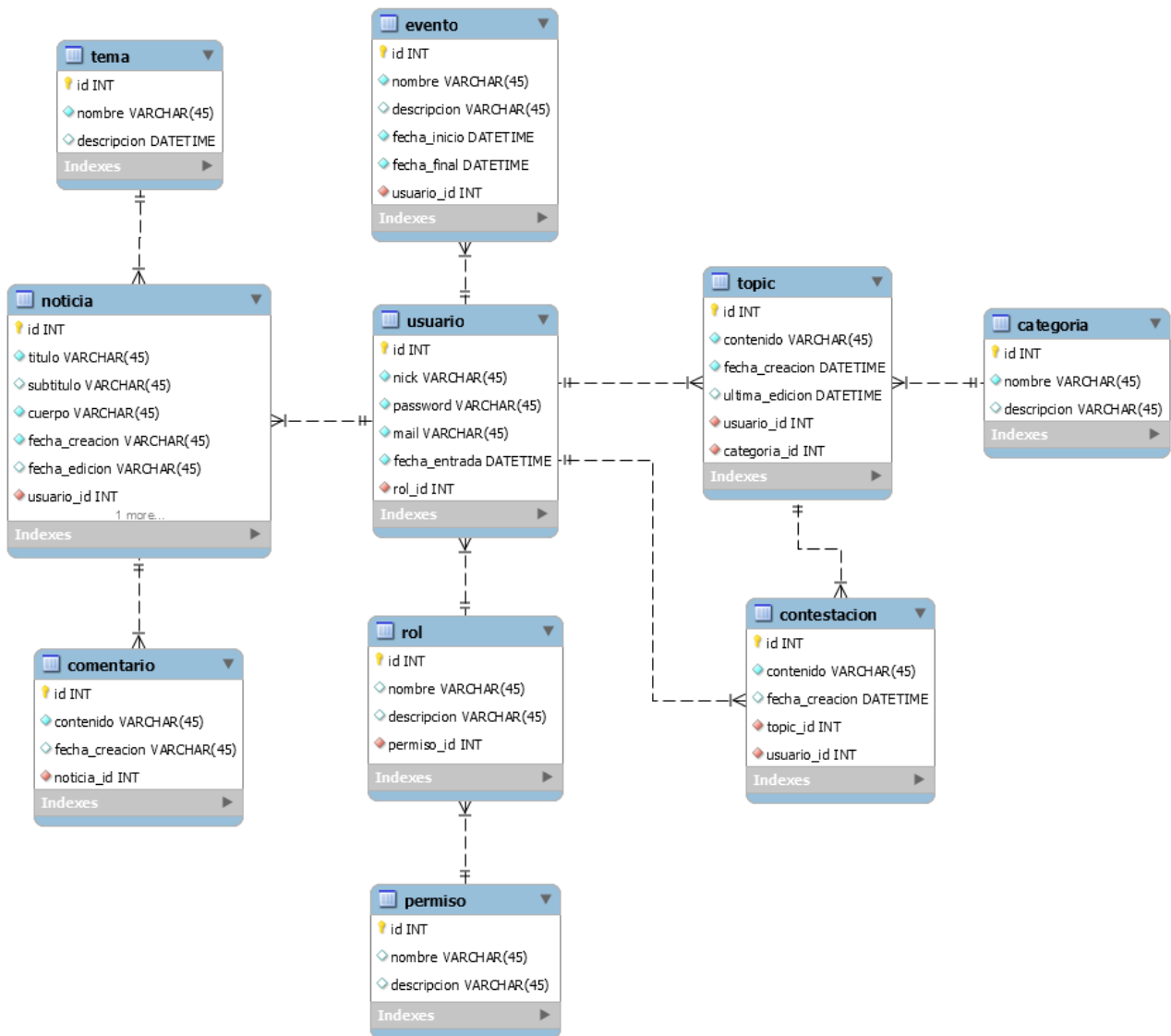
- Tendrá acceso a todas las opciones del redactor

- Podrá editar noticias ajenas o eliminarlas
- Podrá editar eventos ajenos o eliminarlos
- Podrá crear nuevos temas para las noticias
- Podrá eliminar comentarios de las noticias

Opciones disponibles para el moderador:

- Tendrá acceso a todas las opciones del registrado
- Podrá editar posts ajenos o eliminarlos
- Podrá crear nuevas categorías para el foro

Una primera aproximación a la base de datos podría ser:



Con todo esto, ya podemos empezar a pensar en las clases que tendrá el programa:

- Clase de interfaz: Será la clase encargada de mostrar al usuario los elementos de la web con los que puede interactuar, dependiendo el rol que tenga asignado. Deberá estar en estrecho contacto con un sistema de control de usuarios y un sistema de gestión de permisos del que hablaremos a continuación.
- Clase de gestión de usuarios: Cuando un usuario entre a la web esta estará parcialmente oculta, puesto que el usuario será considerado como anónimo y sus privilegios serán escasos. Esta clase se encargará tanto de recibir los usuarios y

contraseñas y comprobar que están en la base de datos como de registrar a los nuevos usuarios que ingresen en la web.

- Mediante el uso de sesiones, se guardarán los datos introducidos para poder navegar entre las diferentes páginas.
 - Se deberá permitir que el usuario guarde su nombre y contraseña para dicha web y que automáticamente se le pueda dar acceso.
- Clase de seguridad: Desde el momento que se permite al usuario introducir texto al servidor (esto es desde que se le permite introducir el propio nick) corremos peligro de que si sus intenciones son maliciosas logre alterar nuestro programa o base de datos. Es por ello que antes de dejar paso las interacciones del usuario a la capa de lógica debemos retenerlas e interrumpir las que puedan dañarnos.
 - Esto no hace referencia solo a tipos de datos no permitidos, si no a líneas de caracteres específicas que el programa no debe leer y mucho menos ejecutar.
 - Se trabajará en Javascript con el uso de patrones cuando estos sean suficientes o complementen a esta capa.
 - Clase de control de permisos: Una vez logueado, el usuario tendrá a su disposición una serie de opciones diferentes según su rol. Estos podrán ser asignados a los usuarios por el administrador en cualquier momento desde su panel de control.
 - Clase CMS: Será la clase encargada de realizar las solicitudes del usuario, o lo que es lo mismo, crear las noticias, eventos, posts en el foro.. siempre y cuando estas solicitudes hayan pasado por la clase de seguridad.
 - Clase de acceso a datos: Dará acceso a toda la aplicación a la base de datos, será el nexo entre la verificación de usuarios, verificación de permisos, entrada de nuevas noticias y posts... y el lugar donde finalmente será almacenados (o donde se realiza dicha búsqueda), la base de datos.
- Características opcionales.- Mejoran el proyecto pero no son necesarias para que la aplicación se considere útil.
 - Creación y gestión de de roles: no es algo necesario, pero no imprescindible para que funcione un CMS. No obstante, es bastante seguro que el administrador pueda crear nuevos roles y asignar los permisos a su antojo.
 - Baneo de usuarios: si bien es cierto que un administrador podría quitarle la categoría de “usuario registrado” a un usuario que estuviera dando problemas, nada le impediría crearse una nueva cuenta. Se debe estudiar la posibilidad de banear la IP de un usuario.
 - Sistema de actualizaciones de estado mediante un observador: Es muy común que cuando alguien contesta a una noticia a la que tu has contestado o que responde a uno de tus posts en el foro te avise mediante una señal gráfica en el perfil de tu usuario y te indique a que url debes dirigirte si deseas ver las actualizaciones. Esto se realizará mediante un escuchador.
 - Entorno en móviles: si bien es cierto que el administrador no se dedicará a gestionar el portal web en una interfaz tan pequeña como la que proporciona el móvil, es cierto que hay cierto tipos de actividades que debería controlar lo antes posible (envío de spam, comentarios insultantes por parte de los usuarios, enlaces a software malicioso...)
 - Algoritmo de posicionamiento de noticias y posts: No todas las noticias son igual de importantes, ni todas las contestaciones en los posts tienen el mismo peso. Hay que ordenar las que el algoritmo considere más importantes en puestos (o tamaños)

visibles para el usuario. Se ha pensado que el algoritmo tendrá en cuenta lo reciente que sea una noticia o post y el número de comentarios o respuestas que esta a generado.

- Periodo de realización

Se calcula que un total de 60 días repartidos entre:

- 2 días dedicados a la estructura HTML
- 5 días dedicados al diseño de plantillas CSS
- 5 días para la codificación javascript
- 2 días para el diseño de la base de datos
- 10 días para la creación del sistema de roles y permisos
- 10 días para la implementación de las noticias
- 5 días para la implementación del foro
- 5 días para la implementación del calendario
- 5 días para programar y testear la capa de seguridad
- 2 días para acoplar la codificación a la capa de presentación
- 2 días para testear la aplicación
- 5 días para corregir errores
- 2 días para ordenar la documentación y redactar el proyecto

3. Entorno de desarrollo

- IDE utilizado, sistema de control de versión, herramientas de documentación, utilidades para la realización de pruebas, software de depuración.

Netbeans IDE 8.0.2: NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java pero que gracias a su modularidad permite no solo programar en otros lenguajes de programación (como en nuestro caso, PHP5) si no que también da la posibilidad de instalar aplicaciones que complementen el desarrollo en ese lenguaje. Las siguientes aplicaciones son módulos que instalaremos en nuestro entorno para documentar y controlar errores.

PHPDocumentor2: Es un generador de documentación de código abierto escrito en PHP. Automáticamente analiza el código fuente PHP y produce la API de lectura y documentación del código fuente en una variedad de formatos. phpDocumentor genera la documentación en base al estándar formal PHPDoc. Es compatible con la documentación del código orientado a objetos y programación procedural, además es capaz de crear documentos HTML, PDF, CHM y formatos Docbook también hay que decir que es capaz de analizar toda la sintaxis de PHP y apoya PHP4 y PHP5.

PHPDebugger: Este es un módulo que se puede incorporar a Netbeans para controlar no solo los errores generados por el programa si no también la ejecución del mismo. Esto es porque permite correr un programa paso a paso (un paso o animación del programa), parar el programa (breaking), es decir, pausar el programa para examinar el estado actual en cierto evento o instrucción especificada por medio de un breakpoint, y el seguimiento de valores de algunas variables.

- Plataforma de seguimiento: GitHub, Bitbucket, servidor Git propio, etc.

GitHub: es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git con las siguientes características:

- Mecanismo de almacenamiento de los elementos que deba gestionar (ej. archivos de texto, imágenes, documentación...).
- Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos).
- Registro histórico de las acciones realizadas con cada elemento o conjunto de

elementos (normalmente pudiendo volver o extraer un estado anterior del producto).

Sistema de gestión de contenidos

Un sistema de gestión de contenidos (en inglés: Content Management System, más conocido por sus siglas CMS) es un programa informático que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás usuarios.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio web. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio web sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior (moderador o administrador) que permite que estos contenidos sean visibles a todo el público (los aprueba).

Historia

Los primeros sistemas de administración de contenidos fueron desarrollados por organizaciones que publicaban una gran cantidad de contenido en Internet, y necesitaban de continuas actualizaciones; como revistas en línea, periódicos y publicaciones corporativas.

En 1995, el sitio de noticias tecnológicas CNET sacó su sistema de administración de documentos y publicación y creó una compañía llamada Vignette, pionero de los sistemas de administración de contenido comerciales.

La evolución de Internet hacia portales con más contenido y la alta participación de los usuarios directamente, a través de blogs y redes sociales, ha convertido a los gestores de contenidos en una herramienta esencial en Internet, tanto para empresas e instituciones como para personas. Los wikis y los sistemas groupware también son considerados CMS.

Hoy en día existen sistemas desarrollados en software libre y software privativo. En ambos casos es necesaria una implantación para adaptar el gestor de contenidos al esquema gráfico y funcionalidades deseadas. Para ciertos gestores existen muchas plantillas disponibles que permite una sencilla implantación de la parte estética por parte de un usuario sin conocimientos de diseño. Los paradigmas de este caso son WordPress y Joomla!, gestores sobre los que hay una gran comunidad de desarrolladores de extensiones (módulos, complementos, plugins, etcétera) y plantillas.

Definición

El gestor de contenido es una aplicación informática usada para crear, editar, gestionar y publicar contenido digital multimedia en diversos formatos. El gestor de contenidos genera páginas web dinámicas interactuando con el servidor web para generar la página web bajo petición del usuario, con el formato predefinido y el contenido extraído de la base de datos del servidor.

Esto permite gestionar, bajo un formato estandarizado, la información del servidor, reduciendo el tamaño de las páginas para descarga y reduciendo el coste de gestión del portal con respecto a un sitio web estático, en el que cada cambio de diseño debe ser realizado en todas las páginas web, de la misma forma que cada vez que se agrega contenido tiene que maquetarse una nueva página HTML y subirla al servidor web.

Otras consideraciones iniciales:

- Entendido como un sistema de soporte a la gestión de contenidos; ya que, en realidad, son las

estrategias de comunicación las que realmente llevan a gestionar contenidos y publicidad de forma efectiva; los sistemas informáticos pueden a lo sumo proporcionar las herramientas necesarias para la publicación en línea, o bien incluir servicios de soporte a la toma de decisiones por lo que a la gestión de contenidos se refiere.

- El gestor de contenidos se aplica generalmente para referirse a sistemas de publicación, pudiendo subestimarse las funcionalidades de soporte y mantenimiento, en detrimento de las funcionalidades relacionadas con la optimización de los tiempos de publicación. La correcta implantación del sistema, con arreglo a las necesidades del cliente es necesaria, y es necesario entender el proyecto de un portal web en el seno de un proyecto de comunicación estructurado y bien planteado.
- La elección de la plataforma correcta será vital para alcanzar los objetivos del cliente, ya que existen particularidades diferenciales tanto en su adaptabilidad a esquemas gráficos como la posible integrabilidad de funcionalidades y extensiones adicionales.
- El posicionamiento en buscadores está relacionado con el volumen de contenidos de un portal y con la forma en la que éste se presenta. Es importante tener eso en cuenta para la estructura del portal para garantizar un correcto posicionamiento orgánico.

Funcionamiento

Un sistema de administración de contenidos siempre funciona en el servidor web en el que esté alojado el portal. El acceso al gestor se realiza generalmente a través del navegador web, y se puede requerir el uso de FTP para subir contenido.

Cuando un usuario accede a una URL, se ejecuta en el servidor esa llamada, se selecciona el esquema gráfico y se introducen los datos que correspondan de la base de datos. La página se genera dinámicamente para ese usuario, el código HTML final se genera en esa llamada. Normalmente se predefinen en el gestor varios formatos de presentación de contenido para darle la flexibilidad a la hora de crear nuevos apartados e informaciones.

Gestión de usuarios

Dependiendo de la plataforma elegida se podrán escoger diferentes niveles de acceso para los usuarios; yendo desde el administrador del portal hasta el usuario sin permiso de edición, o creador de contenido. Dependiendo de la aplicación podrá haber varios permisos intermedios que permitan la edición del contenido, la supervisión y reedición del contenido de otros usuarios, etcétera.

El sistema de gestión de contenidos controla y ayuda a manejar cada paso de este proceso, incluyendo las labores técnicas de publicar los documentos a uno o más sitios. En muchos sitios con estos sistemas una sola persona hace el papel de creador y editor, como por ejemplo en los blogs personales.

Tipos de gestores de contenidos

Los gestores de contenido se pueden clasificar según diferentes criterios:
Por sus características

- Según el lenguaje de programación empleado, por ejemplo:
 - Active Server Pages
 - Java
 - PHP
 - ASP.NET
 - Ruby On Rails
 - Python
 - PERL
- Según la licencia:

- Código abierto
- Software propietario
- Por su uso y funcionalidad
 - Blogs: para páginas personales
 - Foros: para compartir opiniones
 - Wikis: para el desarrollo colaborativo
 - Enseñanza electrónica: plataforma para contenidos de enseñanza en línea
 - Comercio electrónico: plataforma de gestión de usuarios, catálogo, compras y pagos
 - Publicaciones digitales
 - Difusión de contenido multimedia
 - Propósito general
 - Aplicación móvil: plataformas de gestión de aplicaciones móviles.

Ventajas y oportunidades

El gestor de contenidos facilita el acceso a la publicación de contenidos a un rango mayor de usuarios. Permite que sin conocimientos de programación ni maquetación cualquier usuario pueda añadir contenido en el portal web.

Además permite la gestión dinámica de usuarios y permisos, la colaboración de varios usuarios en el mismo trabajo, la interacción mediante herramientas de comunicación. Los costes de gestión de la información son mucho menores ya que se elimina un eslabón de la cadena de publicación, el maquetador. La maquetación es hecha al inicio del proceso de implantación del gestor de contenidos.

La actualización, backup y reestructuración del portal son mucho más sencillas al tener todos los datos vitales del portal, los contenidos, en una base de datos estructurada en el servidor.

El uso de frameworks

1. Introducción

En el mundo de la programación en equipo, es muy importante el concretar ciertos aspectos ajenos a la propia aplicación pero muy necesarios para el correcto desarrollo del mismo. Dichos aspectos están relacionados con la comunicación que mantendrán los diferentes miembros del equipo.

Entendemos por comunicación todas aquellas acciones que conllevan la interacción entre dos o mas programadores, esto engloba desde algo tan superfluo como el programa elegido de comunicación a distancia hasta algo tan necesario como ponerse de acuerdo en el framework (o frameworks) con los que se trabajarán.

2. ¿Qué es un Frameworks?

La palabra inglesa "framework" (infraestructura, almacén, marco) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Los frameworks tienen como objetivo principal ofrecer una funcionalidad definida, auto contenida, siendo contruidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad, se construyen piezas, objetos, llamados objetos calientes, que vinculan las necesidades del sistema con la funcionalidad que este presta. Esta funcionalidad, está constituida por objetos llamados fríos, que sufren poco o ningún cambio en la vida del framework, permitiendo la portabilidad entre distintos sistemas. Frameworks conocidos que se pueden mencionar por ejemplo son Spring Framework, Hibernate, donde lo esencial para ser denominados frameworks es estar contruidos por objetos casi estáticos con funcionalidad definida a nivel grupo de objetos y no como parte constitutiva de estos, por ejemplo en sus métodos, en cuyo caso se habla de un API o librería. Algunas características notables que se pueden observar:

- La inversión de control: en un framework, a diferencia de las bibliotecas, el flujo de control no es dictado por el programa que llama, sino por el mismo.
- La funcionalidad o comportamiento predeterminado: un marco tiene un comportamiento predeterminado. Este comportamiento por defecto debe ser un comportamiento útil, definido e identificable.
- Su extensibilidad: un marco puede ser ampliado para proporcionar una funcionalidad específica. El frame, en general, no se supone que deba ser modificado, excepto en cuanto a extensibilidad. Los usuarios pueden ampliar sus características, pero no deben

ni necesitan modificar su código.

3. Control de versiones

- **Definición:**

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones o VCS (del inglés Version Control System). Estos sistemas facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico). Ejemplos de este tipo de herramientas son entre otros: CVS, Subversion, SourceSafe, ClearCase, Darcs, Bazaar, Plastic SCM, Git, Mercurial, Perforce, Fossil SCM, Team Foundation Server.

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente dando lugar a los sistemas de control de código fuente o SCM (siglas del inglés Source Code Management). Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etc.

- **Características:**

- Mecanismo de almacenamiento de los elementos que deba gestionar (ej. archivos de texto, imágenes, documentación...).
- Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos).
- Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto).
- Aunque no es estrictamente necesario, suele ser muy útil la generación de informes con los cambios introducidos entre dos versiones, informes de estado, marcado con nombre identificativo de la versión de un conjunto de ficheros, etc.

- **Terminología:**

- Repositorio: El repositorio es el lugar en el que se almacenan los datos actualizados e históricos de cambios, a menudo en un servidor. A veces se le denomina depósito o depot. Puede ser un sistema de archivos en un disco duro, un banco de datos, etc..
- Módulo: Conjunto de directorios y/o archivos dentro del repositorio que pertenecen a un proyecto común.
- Revisión ("version"): Una revisión es una versión determinada de la información que se gestiona. Hay sistemas que identifican las revisiones con un contador (Ej. subversion). Hay otros sistemas que identifican las revisiones mediante un código de detección de modificaciones (Ej. Git usa SHA1). A la última versión se le suele identificar de forma especial con el nombre de HEAD. Para marcar una revisión concreta se usan los rótulos o tags.
- Rotular ("tag"): Darle a alguna versión de cada uno de los ficheros del módulo en desarrollo en un momento preciso un nombre común ("etiqueta" o "rótulo") para asegurarse de reencontrar ese estado de desarrollo posteriormente bajo ese nombre. En la práctica se rotula a todos los archivos en un momento determinado. Para eso el módulo se "congela" durante el rotulado para imponer una versión coherente. Pero

bajo ciertas circunstancias puede ser necesario utilizar versiones de algunos ficheros que no coinciden temporalmente con las de los otros ficheros del módulo. Los tags permiten identificar de forma fácil revisiones importantes en el proyecto. Por ejemplo se suelen usar tags para identificar el contenido de las versiones publicadas del proyecto. En algunos sistemas se considera un tag como una rama en la que los ficheros no evolucionan, están congelados.

- Línea base ("baseline"): Una revisión aprobada de un documento o fichero fuente, a partir del cual se pueden realizar cambios subsiguientes.
- Abrir rama ("branch") o ramificar: Un módulo puede ser branched o bifurcado en un instante de tiempo de forma que, desde ese momento en adelante se tienen dos copias (ramas) que evolucionan de forma independiente siguiendo su propia línea de desarrollo. El módulo tiene entonces 2 (o más) "ramas". La ventaja es que se puede hacer un "merge" de las modificaciones de ambas ramas, posibilitando la creación de "ramas de prueba" que contengan código para evaluación, si se decide que las modificaciones realizadas en la "rama de prueba" sean preservadas, se hace un "merge" con la rama principal. Son motivos habituales para la creación de ramas la creación de nuevas funcionalidades o la corrección de errores.
- Desplegar ("Check-out", "checkout", "co"): Un despliegue crea una copia de trabajo local desde el repositorio. Se puede especificar una revisión concreta, y predeterminadamente se suele obtener la última.
- "Publicar" o "Enviar" ("commit", "check-in", "ci", "install", "submit"): Un commit sucede cuando una copia de los cambios hechos a una copia local es escrita o integrada sobre el repositorio.
- Conflicto: El sistema es incapaz de fusionar los cambios. El usuario y debe resolver el conflicto combinando los cambios, o eligiendo uno de ellos para descartar el otro.
- Resolver: El acto de la intervención del usuario para atender un conflicto entre diferentes cambios al mismo archivo.
- Cambio ("change", "diff", "delta"): Un cambio representa una modificación específica a un archivo bajo control de versiones. La granularidad de la modificación considerada un cambio varía entre diferentes sistemas de control de versiones.
- Lista de cambios ("changelist", "change set", "patch"): En muchos sistemas de control de versiones con commits multi-cambio atómicos, una lista de cambios identifica el conjunto de cambios hechos en un único commit. Esto también puede representar una vista secuencial del código fuente, permitiendo que el fuente sea examinado a partir de cualquier identificador de lista de cambios particular.
- Exportación ("export"): Una exportación es similar a un check-out, salvo porque crea un árbol de directorios limpio sin los metadatos de control de versiones presentes en la copia de trabajo. Se utiliza a menudo de forma previa a la publicación de los contenidos.
- Importación ("import"): Una importación es la acción de copia un árbol de directorios local (que no es en ese momento una copia de trabajo) en el repositorio por primera vez.
- Integración o fusión ("merge"): Una integración o fusión une dos conjuntos de cambios sobre un fichero o un conjunto de ficheros en una revisión unificada de dicho fichero o ficheros. independiente por un tiempo, y que entonces se haya requerido que fueran fundidos de nuevo en un único trunk unificado.
- Actualización ("sync" o "update"): Una actualización integra los cambios que han sido hechos en el repositorio (por ejemplo por otras personas) en la copia de trabajo local.
- Copia de trabajo ("workspace"): La copia de trabajo es la copia local de los ficheros de un repositorio, en un momento del tiempo o revisión específicos. Todo el trabajo realizado sobre los ficheros en un repositorio se realiza inicialmente sobre una copia de trabajo, de ahí su nombre. Conceptualmente, es un cajón de arena o sandbox.
- Congelar: significa permitir los últimos cambios (commits) para solucionar las fallas a

resolver en una entrega (release) y suspender cualquier otro cambio antes de una entrega, con el fin de obtener una versión consistente. Si no se congela el repositorio, un desarrollador podría comenzar a resolver una falla cuya resolución no está prevista y cuya solución dé lugar a efectos colaterales imprevistos.

- **Arquitecturas de almacenaminto**

- Centralizados: existe un repositorio centralizado de todo el código, del cual es responsable un único usuario (o conjunto de ellos). Se facilitan las tareas administrativas a cambio de reducir flexibilidad, pues todas las decisiones fuertes (como crear una nueva rama) necesitan la aprobación del responsable. Algunos ejemplos son CVS, Subversion o Team Foundation Server.
- Distribuidos: cada usuario tiene su propio repositorio. Los distintos repositorios pueden intercambiar y mezclar revisiones entre ellos. Es frecuente el uso de un repositorio, que está normalmente disponible, que sirve de punto de sincronización de los distintos repositorios locales. Ejemplos: Git y Mercurial.
- Ventajas de sistemas distribuidos:
 - Necesita menos veces estar conectado a la red para hacer operaciones. Esto produce una mayor autonomía y una mayor rapidez.
 - Aunque se caiga el repositorio remoto la gente puede seguir trabajando
 - Al hacer los distintos repositorio una réplica local de la información de los repositorios remotos a los que se conectan, la información está muy replicada y por tanto el sistema tiene menos problemas en recuperarse si por ejemplo se quema la máquina que tiene el repositorio remoto. Por tanto hay menos necesidad de backups. Sin embargo, los backups siguen siendo necesarios para resolver situaciones en las que cierta información todavía no haya sido replicada.
 - Permite mantener repositorios centrales más limpios en el sentido de que un usuario puede decidir que ciertos cambios realizados por él en el repositorio local, no son relevantes para el resto de usuarios y por tanto no permite que esa información sea accesible de forma pública. Por ejemplo es muy útil se pueden tener versiones inestables o en proceso de codificación o también tags propios del usuario.
 - El servidor remoto requiere menos recursos que los que necesitaría un servidor centralizado ya que gran parte del trabajo lo realizan los repositorios locales.
 - Al ser los sistemas distribuidos más recientes que los sistemas centralizados, y al tener más flexibilidad por tener un repositorio local y otro/s remotos, estos sistemas han sido diseñados para hacer fácil el uso de ramas (creación, evolución y fusión) y poder aprovechar al máximo su potencial. Por ejemplo se pueden crear ramas en el repositorio remoto para corregir errores o crear funcionalidades nuevas. Pero también se pueden crear ramas en los repositorio locales para que los usuarios puedan hacer pruebas y dependiendo de los resultados fusionarlos con el desarrollo principal o no. Las ramas dan una gran flexibilidad en la forma de trabajo.
- Ventajas de sistemas centralizados:
 - En los sistemas distribuidos hay menos control a la hora de trabajar en equipo ya que no se tiene una versión centralizada de todo lo que se está haciendo en el proyecto.
 - En los sistemas centralizados las versiones vienen identificadas por un número de versión. Sin embargo en los sistemas de control de versiones distribuidos no hay números de versión, ya que cada repositorio tendría sus propios números de revisión dependiendo de los cambios. En lugar de eso cada versión tiene un identificador al que se le puede asociar una etiqueta (tag).

4. GitHub

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

5. Idioma, nomenclatura y documentación

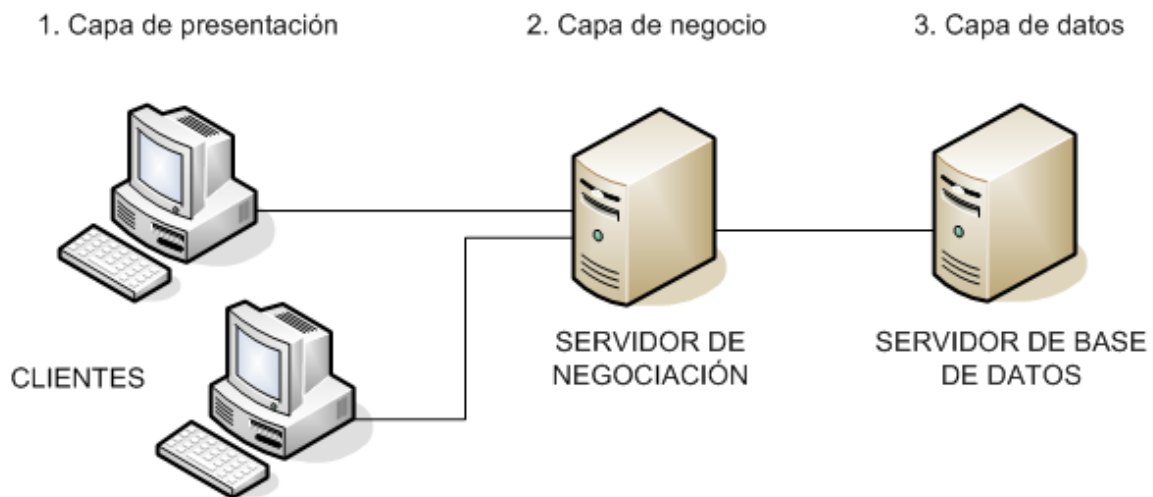
Una vez dicho esto, ya solo nos queda por aclarar que como miembros de un supuesto grupo de programadores, deberemos también ponernos de acuerdo en el idioma en que comentaremos el código (puesto que hay que prever si dicho código será leído por personas que no entiendan el castellano) la nomenclatura de las variables (pudiendo estar escritas en pascal case, camel case, kebab case...) o la manera en que se va a documentar el proyecto. En nuestro caso hemos elegido el phpdocumentor2 como plugin para documentar nuestra aplicación.

PhpDocumentor es un generador de documentación de código abierto escrito en PHP. Automáticamente analiza el código fuente PHP y produce la API de lectura y documentación del código fuente en una variedad de formatos. phpDocumentor genera la documentación en base al estándar formal PHPDoc. Es compatible con la documentación del código orientado a objetos y programación procedural, además es capaz de crear documentos HTML, PDF, CHM y formatos Docbook. Se puede utilizar desde la línea de comandos o mediante una interfaz web. Tiene soporte para la vinculación entre la documentación, la incorporación de documentos a nivel de usuario como tutoriales, y la creación de código fuente resaltado con referencias cruzadas a la documentación en general de PHP. phpDocumentor es capaz de analizar toda la sintaxis de PHP y apoya PHP4 y PHP5. Se trata de un proyecto de código abierto y se distribuye bajo la licencia LGPL.

Modelo Vista Controlador

1. Introducción

A parte de lo ya comentado sobre el trabajo en equipo. Una de las características más importantes de las webs dinámicas es su arquitectura. Al recibir feedback por parte del usuario, se hace imprescindible el controlar la información recibida y mostrada. Para ello, es necesario crear una capa intermedia entre ambas partes que controle el flujo de datos.



La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

En el diseño de sistemas informáticos actual se suelen usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

2. Patrones de diseño de software

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. En nuestro caso en particular, para resolver nuestro problema de arquitectura en capas, lo que haremos será elegir un patrón de arquitectura llamado Modelo Vista Controlador, un patrón que explicaremos más adelante.

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber

comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

En 1979 el arquitecto Christopher Alexander aportó al mundo de la arquitectura el libro *The Timeless Way of Building*; en él proponía el aprendizaje y uso de una serie de patrones para la construcción de edificios de una mayor calidad, en la que esa mayor calidad se refería a la arquitectura antigua y la menor calidad correspondía a la arquitectura moderna, que el romper con la arquitectura antigua había perdido esa conexión con lo que las personas consideraban que era calidad.

En palabras de este autor, "Cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar esta solución un millón de veces más adelante sin tener que volver a pensarla otra vez."

Los patrones que Christopher Alexander y sus colegas definieron, publicados en un volumen denominado *A Pattern Language*, son un intento de formalizar y plasmar de una forma práctica generaciones de conocimiento arquitectónico. Los patrones no son principios abstractos que requieran su redescubrimiento para obtener una aplicación satisfactoria, ni son específicos a una situación particular o cultural; son algo intermedio. Un patrón define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones. Dentro de las soluciones de Christopher Alexander se encuentran cómo se deben diseñar ciudades y dónde deben ir las perillas de las puertas.

Más tarde, en 1987, Ward Cunningham y Kent Beck, sobrepasados por el pobre entrenamiento que recibían los nuevos programadores en orientación a objetos, se preguntaban cómo se podían capturar las buenas ideas para luego de alguna manera traspasarlas a los nuevos programadores recién instruidos en herencia y polimorfismo. Leyendo a Alexander se dieron cuenta del paralelo que existía entre la buena arquitectura propuesta por Alexander y la buena arquitectura OO, de modo que usaron varias ideas de Alexander para desarrollar cinco patrones de interacción hombre-ordenador (HCI) y publicaron un artículo en OOPSLA-87 titulado *Using Pattern Languages for OO Programs*.

No obstante, no fue hasta principios de la década de 1990 cuando los patrones de diseño tuvieron un gran éxito en el mundo de la informática a partir de la publicación del libro *Design Patterns* escrito por el grupo Gang of Four (GoF) compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, en el que se recogían 23 patrones de diseño comunes.

3. Descripción

En líneas generales, MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, si lo prefieres en inglés. En este artículo estudiaremos con detalle estos conceptos, así como las ventajas de ponerlos en marcha cuando desarrollamos.

MVC es un "invento" que ya tiene varias décadas y fue presentado incluso antes de la

aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web.

4. ¿Por qué elegir Modelo Vista Controlador?

- Aunque no tenga nada que ver, comencemos con algo tan sencillo como son el HTML y las CSS. Al principio, en el HTML se mezclaba tanto el contenido como la presentación. Es decir, en el propio HTML tenemos etiquetas como "font" que sirven para definir las características de una fuente, o atributos como "bgcolor" que definen el color de un fondo. El resultado es que tanto el contenido como la presentación estaban juntos y si algún día pretendíamos cambiar la forma con la que se mostraba una página, estábamos obligados a cambiar cada uno de los archivos HTML que componen una web, tocando todas y cada una de las etiquetas que hay en el documento. Con el tiempo se observó que eso no era práctico y se creó el lenguaje CSS, en el que se separó la responsabilidad de aplicar el formato de una web.
- Al escribir programas en lenguajes como PHP, cualquiera de nosotros comienza mezclando tanto el código PHP como el código HTML (e incluso el Javascript) en el mismo archivo. Esto produce lo que se denomina el "Código Espagueti". Si algún día pretendemos cambiar el modo en cómo queremos que se muestre el contenido, estamos obligados a repasar todas y cada una de las páginas que tiene nuestro proyecto. Sería mucho más útil que el HTML estuviera separado del PHP.
- Si queremos que en un equipo intervengan perfiles distintos de profesionales y trabajen de manera autónoma, como diseñadores o programadores, ambos tienen que tocar los mismos archivos y el diseñador se tiene necesariamente que relacionar con mucho código en un lenguaje de programación que puede no serle familiar, siendo que a éste quizás solo le interesan los bloques donde hay HTML. De nuevo, sería mucho más fácil la separación del código.
- Durante la manipulación de datos en una aplicación es posible que estemos accediendo a los mismos datos en lugares distintos. Por ejemplo, podemos acceder a los datos de un artículo desde la página donde se muestra éste, la página donde se listan los artículos de un manual o la página de backend donde se administran los artículos de un sitio web. Si un día cambiamos los datos de los artículos (alteramos la tabla para añadir nuevos campos o cambiar los existentes porque las necesidades de nuestros artículos varían), estamos obligados a cambiar, página a página, todos los lugares donde se consumían datos de los artículos. Además, si tenemos el código de acceso a datos disperso por decenas de lugares, es posible que estemos repitiendo las mismas sentencias de acceso a esos datos y por tanto no estamos reutilizando código.

5. Partes del Modelo Vista Controlador

- **Modelo:**

Es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes selects, updates, inserts, etc.

No obstante, cabe mencionar que cuando se trabaja con MCV lo habitual también es utilizar otras librerías como PDO o algún ORM como Doctrine, que nos permiten trabajar con abstracción de bases de datos y persistencia en objetos. Por ello, en vez de

usar directamente sentencias SQL, que suelen depender del motor de base de datos con el que se esté trabajando, se utiliza un dialecto de acceso a datos basado en clases y objetos.

- **Vista**

Las vistas, como su nombre nos hace entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

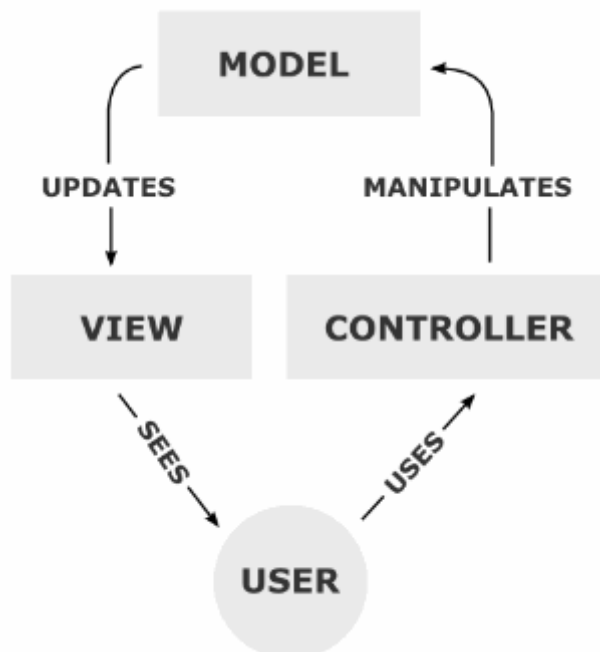
En la vista generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas se generará la salida, tal como nuestra aplicación requiera.

- **Controlador**

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

En realidad es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

6. Flujo de la información



- El usuario realiza una solicitud a nuestro sitio web. Generalmente estará desencadenada por acceder a una página de nuestro sitio. Esa solicitud le llega al controlador.
- El controlador comunica tanto con modelos como con vistas. A los modelos les solicita datos o les manda realizar actualizaciones de los datos. A las vistas les solicita la salida

correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.

- Para producir la salida, en ocasiones las vistas pueden solicitar más información a los modelos. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación; por eso esa flecha la hemos coloreado de otro color.
- Las vistas envían al usuario la salida. Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente, por eso he puesto la flecha en otro color.

Bibliografía

B.O.E.

- <https://www.boe.es/boe/dias/2011/07/30/pdfs/BOE-A-2011-13118.pdf>
- <https://www.boe.es/boe/dias/2010/11/11/pdfs/BOE-A-2010-17329.pdf>

Wikipedia:

- https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos
- https://en.wikipedia.org/wiki/Content_management_system
- <https://es.wikipedia.org/wiki/PhpDocumentor>
- https://es.wikipedia.org/wiki/Control_de_versiones
- <https://es.wikipedia.org/wiki/Framework>
- <https://es.wikipedia.org/wiki/GitHub>
- https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas
- https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o

MVC

- <http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>
- <http://www.desarrolloweb.com/articulos/que-es-mvc.html>
- <http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>

CMS

- <http://blogspot.es/>
- <https://es.wordpress.com/>
- <https://www.joomla.org/>
- <http://www.1and1.es/>
- <https://www.one.com/es/>

Programación:

- <http://www.w3schools.com/>
- <http://php.net/docs.php>
- <http://www.phpdoc.org/docs/latest/index.html>
- <https://help.github.com/>