

PFM-F0. INTRODUCCIÓ

Diseño de Interfaces WEB
CFGS DAW

Alfredo Oltra
alfredo.oltra@ceedcv.es

2020/2021

Versión:200918.1926

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

⚡ Atención

🔊 Interesante

ÍNDICE DE CONTENIDO

1. Introducción.....	4
2. Planificación del curso.....	4
2.1 Puntos de control.....	5
3. Conocimientos previos.....	6
3.1 Git.....	6
3.2 Javascript.....	6
3.3 Otros conceptos.....	6
4. Evaluación.....	7
4.1 Entrega.....	7
4.2 Evaluación.....	7
4.2.1 Rúbrica de evaluación.....	9
5. Licencias.....	10

PFM-F0. INTRODUCCIÓN

1. INTRODUCCIÓN

El proceso de evaluación del curso va a consistir en lo que a partir de ahora denominaremos *Proyecto Fin de Módulo* o, para abreviar, PFM. La idea es dejar de lado las limitaciones que tenemos con un examen escrito y poder valorar de una mejor manera las habilidades reales del alumno en lo referente al módulo de Diseño de Interfaces Web.

Además con el trabajo no se pretende únicamente evaluar, también que sirva como punto de partida para seguir aprendiendo no sólo técnicas relacionadas con el módulo sino también con temas transversales como la metodología a seguir.

El problema propuesto no es trivial, pero eso no significa que sea difícil. Lo que hay que entender es que el trabajo a realizar para buscar el aprobado no se realiza en tres o cuatro tardes. Es necesario comprender lo que se pide, diseñar en consecuencia, codificar, probar, etc... Es muy importante que tengas en cuenta que el PFM es la forma de evaluar todo el curso, por lo que no debes pensarlo como una actividad evaluable. Es la unión de las evaluables y de los exámenes de la primera y segunda evaluación, por lo que la exigencia va a ser importante, al igual que el tiempo a dedicar.

Por todo esto es fundamental una buena planificación y el buen uso de los recursos disponibles, que empiezan con esta documentación. Es muy importante que leas en profundidad, tanto este como el resto de documentos que se irán publicando a lo largo del curso y que utilices los recursos habituales para poder compartir información, problemas o posibles soluciones.

2. PLANIFICACIÓN DEL CURSO

Para intentar que el trabajo sea escalonado, la planificación del proyecto se divide en 7 fases, 4 en la primera evaluación y 3 en la segunda. Cada una de ellas irá paralela a los contenidos que se vayan impartiendo en el curso, de manera que poco a poco se construya el resultado final.

⚡ La única forma de seguir la evaluación continua del módulo es mediante la realización del PFM. Aquellos que no quieran evaluarse a través del proyecto pueden ir al examen de la convocatoria ordinaria o de la extraordinaria.

En el aula virtual irán apareciendo los diferentes enunciados de cada fase. Salvo la primera y la quinta, la apertura de cada una de ellas dependerá de la finalización de la fase anterior, es decir, no se mostrará hasta que la fase anterior haya sido entregada. Por su parte, la primera y a la quinta dependerán también del momento programado de publicación. Dichas fechas, que aparecen en el calendario de la plataforma a modo de eventos, son:

Fase	Fecha
1	5 de octubre de 2020 a las 0:00
5	4 de diciembre de 2020 a las 0:00

🔊 Esta planificación da libertad al alumno para ir a su ritmo en cada evaluación, pudiendo avanzar incluso a mayor ritmo de los contenidos impartidos en el curso.

El proyecto deberá ser controlado mediante *git* y alojado en *GitLab*. Para poder coordinarlo es necesario que rellenes la encuesta que aparece en el aula, en el apartado *Proyecto Fin de Módulo*, indicando cuál es tu cuenta de usuario. Si tuvieras cualquier problema con la cuenta, coméntalo con el profesor lo más urgentemente posible.

⚡ El alumno que no proporcione su cuenta no podrá ser evaluado y por lo tanto se le asignará un cero como nota de PFM.

2.1 Puntos de control

El objetivo de esta planificación es doble. Como se ha comentado en la introducción, el PFM necesita de trabajo continuado en el tiempo (no picos de trabajo en los fechas cercanas a la entrega). Además resulta muy importante que el alumno sepa en todo momento que va por el camino adecuado, y que no se está exigiendo demasiado (o demasiado poco).

En esa línea también se plantea también la existencia de unos puntos de control, revisiones puntuales del proyecto por parte del profesor para comprobar que el desarrollo del trabajo es el adecuado. Ni el cumplimiento de las fechas de entrega tope de las fases ni la asistencia a los puntos de control son obligatorias (es decir no se suspenderá a nadie si no las cumple), pero si serán tenidas en cuenta a la hora de evaluar.

⚡ El único incumplimiento de plazo que si implicaría un suspenso será el de la entrega final del proyecto, tanto en convocatoria de primera o segunda evaluación (fase 4 y 7 respectivamente) como en ordinaria.

La forma de cumplimentar los puntos de control será o vía tutoría individual (presencial o a distancia) o mediante acceso al repositorio del alumno. En cualquier caso el profesor no actuará de oficio, será siempre el alumno el que deberá indicarle que quiere pasar el punto de control¹.

¹ A efectos de simplificar el concepto, se podría decir que un punto de control es una tutoría individual centrada en el seguimiento del PFM y de la que el profesor guarda un registro.

3. CONOCIMIENTOS PREVIOS


Además de los conocimientos impartidos durante el curso, son necesarios unos conocimientos previos relativos tanto en el campo conceptual como en el procedimental. A continuación se enumeran algunos de los más importantes.

3.1 Git

En este caso su uso es necesario para poder gestionar correctamente el día a día del PFM. Aunque en la propia documentación del proyecto se irán comentando los aspectos más novedosos que se vayan a utilizar, es necesario disponer de cierta soltura para poder finalizar el trabajo correctamente.

Recuerda la filosofía de uso de un control de versiones:

- Úsalo siempre, siempre, siempre. Aunque programes sólo, aunque sea una práctica de 10 líneas. Tu primer paso cuando empieces un proyecto es crear la carpeta. El segundo entrar en ella y hacer un *git init*. El tercero abrir la herramienta de trabajo *netbeans*, *brackets*, *sublime* o la que sea. Ahora te parece un rollo, dentro de 1 año lo harás de manera natural.
- Haz *commits* sin miedo. Los *commits* están para usarse. En general cuantos más hagas mejor: más posibilidades tendrás de poder volver a otro punto del código, menos de tener conflictos cuando estés trabajando con otros programadores, etc. Cada vez que hagas una pequeña tarea que te has propuesto, *commit*, cada vez que corrijas una error (aunque sólo haya sido cambiar dos letras), *commit*...

 Es muy recomendable que en cuanto empieces el proyecto me des permisos de lectura (no al final) y que hagas *push* al servidor lo más a menudo posible para que yo pueda ir viendo el proceso de creación, el histórico de los cambios. Si sólo subes un *commit* y encima sólo me dejas verlo al final no te va a aportar nada a ti, ni tampoco a mi a la hora de corregirlo.

El uso de un sistema de control de versiones y en concreto de Git se imparte en el módulo de Entornos de Desarrollo de primero.

3.2 Javascript

Parte de los contenidos de la segunda evaluación se basan en el uso de javascript, por lo que un nivel medio de este lenguaje es necesario para poder hacer las implementaciones que se solicitan. Los conceptos necesarios de javascript se imparten en el módulo de Desarrollo Web en Entorno Cliente de segundo. Si ya has cursado ese módulo o lo estás cursando este año no tendrás ningún problema (los contenidos de ambos módulos se han sincronizado).

3.3 Otros conceptos


Aunque no son vitales para el desarrollo del PFM, conocimientos previos de *Docker*, *GNU/Linux*, *API REST* y *Web Sockets* ayudarán a un mejor entendimiento del proceso de creación.

Por otro lado puede ser interesante tener conocimientos sobre despliegue y *CI/CD*.

4. EVALUACIÓN

4.1 Entrega


Como se ha comentado anteriormente, el proyecto deberá ser controlado mediante *git* y alojado en *GitLab*. Los proyectos tienen que ser privados y han de incluir al usuario *alfredo.ptcf* como usuario con permisos de lectura. Para ello, desde la página de *GitLab*, hay que acceder al repo creado y seleccionar desde la barra lateral de la izquierda el menú de *Settings*. Desde ahí, en la opción *Select members to invite* escribir *@alfredo.ptcf* y asignarle rol de *Reporter*.

 Todas las fases, excepto la primera, serán entregadas vía *GitLab*

Las fases 1, 2, 3, 5 y 6 tendrán asignada una fecha de entrega recomendada. Estas fechas estarán ubicadas para que el alumno mantenga un flujo de trabajo uniforme, pero no son de obligado cumplimiento.


Las fechas de las fases 4 y 7 sí son importantes ya que definen la nota de la primera evaluación y final de curso respectivamente. Esas fechas, junto con la de la entrega para la convocatoria ordinaria y extraordinaria para aquellos que no aprueben por evaluaciones el proyecto aparecen indicadas en la siguiente tabla.

Fase	Evaluación	Fecha límite
4	Primera evaluación	7 de diciembre de 2020 a las 23:55
7	Segunda evaluación	22 de febrero de 2021 a las 23:55
-	Convocatoria ordinaria	8 de marzo de 2021 a las 23:55
-	Convocatoria extraordinaria	25 de mayo de 2021 a las 23:55

 Los días siguientes a la entregas, a las 0:05 se lanzará un script de clonación de todos los proyectos con relación a la cuenta asignada. Si el proyecto no es clonado, se dará como no entregado y se calificará con un cero (0).

4.2 Evaluación

El trabajo tiene carácter individual e intransferible. Para considerar el proyecto realizado, no basta únicamente con entregar el código: el alumno ha de ser capaz de defender su ejercicio a petición del profesor y ser capaz de realizar pequeñas modificaciones relacionadas con el mismo, con el fin de demostrar la adquisición de conocimiento y evitar cualquier sospecha de copia.

 De hecho, *con total seguridad*, varios alumnos serán llamados a defender en privado el trabajo con el profesor. Esa defensa se realizará de manera presencial y hará las veces del examen. La no asistencia a esa defensa implicará el suspenso del PFM y por lo tanto de la evaluación o de la convocatoria correspondiente.

La fechas disponibles para la posible defensa del PFM por parte del alumno serán publicadas con antelación a cada convocatoria de exámenes. El horario se determinará con el alumno para facilitar su asistencia.

⚡ La copia se castiga con el suspenso del módulo completo.

Por lo que respecta al sistema de evaluación:

- Sólo pueden acceder a la evaluación por proyectos aquellos alumnos que presenten el proyecto al menos en una de las evaluaciones o, lo que es lo mismo, que hayan entregado antes de las fechas límite de las evaluaciones alguna de las fases. En cualquier otro caso el alumno tendrá que hacer un examen de todo el contenido del curso en la evaluación ordinaria o extraordinaria.
- El sistema de evaluación será continuo, de manera que un alumno que suspenda la primera evaluación podrá aprobar el curso aprobando la segunda.
- En caso de que el alumno suspendiese el PFM en la segunda evaluación, podrá repetir la entrega del proyecto en la evaluación ordinaria.
- La entrega de las fases deberá realizarse de manera secuencial. Una fase posterior no será tomada en cuenta si una fase previa no ha sido entregada. Por ejemplo, aunque esté entregada, la fase 2 se consideraría como no presentada si la fase 1 no ha sido entregada.
- La evaluación de las fases (excepto la 4 y la 7) se realizará únicamente para dar información al alumno del estado de su trabajo. No tiene un valor cuantitativo ni definitorio en la nota de la evaluación o de la ordinaria. De hecho, las fases se evaluarán de manera cualitativa con los valores:
 - No apto: es necesario hacer cambios importantes para poder considerar la fase superada.
 - Mejorable: aunque en algunos aspectos la entrega es aceptable, en otros no y en general es mejorable y muy recomendable realizar cambios.
 - Apto: el trabajo es suficiente para considerar la fase superada.
 - Buen trabajo: el trabajo realizado es de buena calidad (no implica que no pueda mejorarse).
- En las fases 4 y 7 (las relacionadas con las convocatorias de exámenes) se realizará una valoración global (con nota numérica) de las fases anteriores.
- La nota del proyecto en cada evaluación y, en caso de ser necesaria, de la convocatoria ordinaria vendrá dada por la nota de la fase 4 (primera evaluación) y la fase 7 (segunda evaluación , ordinaria y extraordinaria).
- La nota final del módulo será la nota final del proyecto más la participación en foros. Esta nota puede provenir de la segunda evaluación, de la convocatoria ordinaria o de la extraordinaria.

4.2.1 Rúbrica de evaluación

Diseño (29%)			Funcionalidad (69%)		
Aspecto	Pond.	Valoración	Aspecto	Pond.	Valoración
Planificación diseño	5%	0-10	Errores	16%	0-10
Usabilidad	4%	0-10	Cumplimiento requisitos	16%	0-10
Estética	2%	0-10	Uso correcto de técnicas	10%	0-10
Accesibilidad ²	4%	0-10	Uso variado de técnicas	10%	0-10
Diseño responsable	5%	0-10	Buenas prácticas	5%	0-10
Originalidad	4%	0-10	Documentación del código	5%	0-10
Contenido	5%	0-10	Nuevas especificaciones	5%	0-10
Otros aspectos					
Cumplimiento de los puntos de control				2%	0-3
Defensa (sólo en el caso de aquellos alumnos que tengan que hacerla) ³				---	0-10

² Aunque el *site* no sea completamente accesible, se tendrá en cuenta la inclusión de elementos accesibles

³ En este caso la ponderación no se indica ya que afectará a uno o varios de los aspectos de anteriores.

5. LICENCIAS

Salvo que se indique lo contrario, el proyecto está liberado con licencia GPL – 3.0.

Las imágenes proporcionadas vienen con sus respectivas licencias y créditos que hay que incluir de manera obligatoria en el README.md (apartado licencias) y en la página web.

En cuanto al resto, todas las imágenes, iconos, texturas, etc que se utilicen deben estar correctamente usadas. Por lo tanto, no podrán ser usadas imágenes propietarias y en el caso de licencias CC-* -by el autor deberá aparecer en el README.md del proyecto (apartado recursos), así como en los créditos de la propia página.

En el caso de que sea el alumno el que desarrolle sus propias imágenes deberá indicarlo en el README.md (apartado recursos) , así como indicar el tipo de licencia que usa para ellas.