1. Inicio

Normativa del proyecto

Se deberá desarrollar y desplegar una aplicación web donde se demuestren todos los conocimientos adquiridos a lo largo del Ciclo Formativo. La aplicación debe contar con una parte de administración o gestión a la que sólo se puede acceder mediante el rol correspondiente. Por ejemplo, si la aplicación es un planificador de bodas, se debe implementar también una parte de administración para poder gestionar el contenido: banquetes, regalos, etc. Sea como sea, la aplicación deberá cumplir los requisitos indicados por el profesorado de cada módulo.

Además, se deberán tener en cuenta los siguientes aspectos:

a. El proyecto de Desarrollo de Aplicaciones Web se desarrollará preferentemente en parejas. En el caso de que el alumno quisiera realizar el proyecto de manera individual, deberá comunicar al tutor de grupo de manera formal (correo corporativo, iPasen ...), justificando convenientemente su propuesta, para que el equipo educativo valore su idoneidad.
b. Cada pareja tendrá que trabajar conjuntamente en el proyecto y mantener un detallado diario de las tareas realizadas por cada uno. Igualmente, aquellos que realicen el trabajo de forma individual, también tendrán que detallar las tareas que conllevan a la realización del proyecto.
c. Se creará en un repositorio privado de Github dando permiso al profesorado para que pueda tener acceso en todo momento al desarrollo de los proyectos que deberá contener:

1. Código fuente de la aplicación actualizado semanalmente.
2. Título del proyecto.
3. Nombre de los autores del proyecto.
4. Descripción.
5. Objetivos del proyecto.
6. Tecnologías utilizadas.
7. Esquema E/R de la base de datos.
8. Breve tutorial de uso de la aplicación con capturas de pantalla y/o material multimedia.
9. URL donde se encuentre desplegada la aplicación.
10. URL donde se encuentre desplegada la *landing page*.
11. URL donde se encuentre el diseño de la aplicación.
12. Presentación en formato PDF.
13. Bitácora de tareas realizadas donde se indique la fecha en que se completó cada una de ellas y qué miembro del grupo la llevó a cabo.
14. Bibliografía utilizada (manuales, enlaces a documentación, tutoriales, etc.)
15. URL a un vídeo de 10 minutos máximo, donde los alumnos expongan brevemente su proyecto y muestren su funcionamiento. Es importante hacer una introducción diciendo el nombre de la aplicación y de qué trata en una frase,

antes de pasar a los detalles técnicos. Se puede enseñar código, pero solo fragmentos y si realmente son cosas de interés. El vídeo puede estar colgado en plataformas como Youtube, Vimeo, Dailymotion, etc.

Se valorará positivamente la claridad, estructuración, eficiencia y comentarios del código, así como el uso del inglés en la documentación y/o exposición final.

Anteproyecto

El proyecto se iniciará con la entrega de un anteproyecto que el alumnado deberá presentar obligatoriamente antes del día y hora indicado por el equipo docente. En él se especificará, en términos generales, el nombre de los autores del proyecto, el título, los objetivos, temática, tecnología utilizada tanto en el *backend* como en el *frontend*, etc., teniendo en cuenta los puntos a, c, c.2, c.3, c.4, c.5, c.6 y c.7 indicados en el punto anterior.

Se debe elaborar el Anteproyecto en Notion y añadir un enlace en el README.MD del repositorio de GitHub.

El profesorado valorará la adecuación de la propuesta dando el visto bueno o solicitando al alumnado correcciones y/o mejoras que deberán entregarse en el plazo indicado. El proyecto no deberá comenzarse hasta obtener el visto bueno del equipo docente al anteproyecto presentado.

Revisión (checkpoint)

El equipo docente indicará fecha y hora de realización del *checkpoint*, momento en el que el alumnado que esté realizando su proyecto final del ciclo deberá subir obligatoriamente a Github una primera versión funcional del proyecto, además de:

- Código actualizado.
- Histórico de cambios, bibliografía, etc., todo ello especificado convenientemente en el README
- URL a un vídeo explicativo, de no más de 5 minutos, donde se comente qué características de la aplicación se han desarrollado hasta la fecha, y qué queda por completar. El vídeo podrá estar subido subido a cualquier plataforma de difusión de contenidos como Youtube, Vimeo, etc.

El profesorado revisará los progresos realizados y podrá abrir incidencias en el repositorio que deberán ser completadas por los alumnos antes de continuar con su proyecto.

Exposición

La duración de cada exposición será de 15 minutos (10 de exposición y 5 de preguntas) para los grupos de dos personas. El alumnado que haya decidido realizar el proyecto

sólo tendrá 5 minutos menos de exposición. Se dejará un margen de 5 minutos adicionales entre exposiciones. El alumnado deberá llegar con tiempo suficiente a su presentación. La exposición del proyecto es obligatoria. Si, por causas justificadas, algún alumno/a no llega la hora indicada en el calendario de exposiciones, deberá notificarlo al profesorado y esperar a que se le indique un nuevo día y hora para realizar la exposición.

Se plantea a continuación una guía de ítems que deberían comentarse durante la exposición, teniendo en cuenta que el alumnado deberá ajustarse siempre al tiempo asignado.

- Presentación
- Nombre del proyecto
- Objetivos del proyecto
- Tecnologías utilizadas
- Justificación de las tecnologías (en caso de que no sean las propuestas por el profesorado)
- Reparto de tareas (si es un grupo)
- Breve demostración de uso
- Dificultades encontradas
- Posibles mejoras
- Conclusión
- Despedida

2. Desarrollo web en Entorno Servidor

Utilizando Laravel 11/12, desarrollar una aplicación web de temática libre que contemple, como mínimo, dos perfiles: usuario y administrador. Además, se desarrollará una API REST que proporcione servicios para interactuar con clientes externos.

Tecnologías utilizadas

- Next.js
- CSS
- C#

Requisitos funcionales

Autenticación

- Se deberá implementar un adecuado control de usuarios.

Validación de datos

- Se deberá realizar validaciones de los datos proporcionados por los usuarios a través de formularios, para lo cual deberán utilizarse las reglas proporcionadas por C#y crearse aquellas nuevas reglas que sean necesarias.
- Las vistas mostrarán los mensajes de error convenientemente dependiendo del resultado de la validación de datos.

Migraciones, semilleros y factorías

- La base de datos deberá construirse a través de migraciones y poblarse utilizando semilleros y factorías.
- El esquema E/R deberá estar presente en el repositorio desde la entrega del anteproyecto.

Protección de rutas

- Se deberá gestionar de forma correcta los accesos a zonas privadas a través de *middlewares* y las herramientas ofrece para ello. Se podrán crear *middlewares* adicionales en caso de que sea necesario.

Agrupación y nombrado de rutas

- Se deberán nombrar aquellas rutas que lo necesiten.
- Se organizarán las rutas en grupos según su funcionalidad y/o nivel de acceso.

Vistas y plantillas

- El front del perfil de administración deberá realizarse obligatoriamente utilizando las herramientas que proporciona Laravel para la creación de vistas: vistas con herencia, componentes y slots.
- Se deberán utilizar los mecanismos de paginación.

Extendiendo

- En caso necesario, se crearán y utilizaran macros, helpers y directivas que amplíen las funcionalidades del framework y doten de mayor flexibilidad a controladores y vistas.

Internacionalización

- La aplicación deberá internacionalizarse convenientemente utilizando, el amenos, dos idiomas: español e inglés.

API REST

- Se desarrollará una API REST que dará servicio al *front* de la parte de usuario, que se desarrollará según las especificaciones del módulo de Desarrollo Web en Entorno Cliente.
- Se utilizará autenticación y registro de usuarios.
- Se deberá generar una documentación completa para la API utilizando herramientas como Swagger , definiendo de forma concisa los *endpoints* de la API y sus operaciones asociadas, como por ejemplo:

GET /api/users : Obtener listado de usuarios.
POST /api/users : Crear un nuevo usuario.
GET /api/users/{id} : Obtener detalles de un usuario específico.
PUT /api/users/{id} : Actualizar información de un usuario.
DELETE /api/users/{id}: Eliminar un usuario.

Otras tecnologías de servidor

i se desea utilizar otras tecnologías del lado del servidor, deberán indicarse en el anteproyecto y justificar su utilización de forma apropiada. El profesor del módulo de Desarrollo web en Entorno Servidor realizará una valoración de la propuesta y decidirá si el proyecto podrá desarrollarse empleando las tecnologías indicada; en otro caso, deberá realizarse utilizando Laravel.

3. Despliegue de Aplicaciones Web

El alumnado deberá tener en cuenta los siguientes apartados a la hora de realizar el proyecto:

Requisitos Mínimos

- El despliegue se debe realizar de manera obligatoria en AWS.
    1. Se deberá realizar en al menos una instancia EC2. Alternativamente el alumno puede decidir usar EKS o ECS.
    2. No se puede utilizar AWS Beanstalk o cualquier otro servicio administrado de AWS.
- Al menos se debe tener un servidor web y/o de aplicaciones con todo los recursos integrados (lenguaje de programación, BBDD, servidor web).
    1. Debe tener una IP elástica asignada y ser accesible mediante SSH.
    2. Deberá funcionar en HTTPS.

Para alcanzar una calificación superior en el módulo de Despliegue de Aplicaciones Web se deben implementar alguno de los requisitos adicionales.

Requisitos adicionales

- Adicionalmente se podrá hacer uso de:
    1. Se puede usar RDS para alojar el Servidor de Base de Datos
    2. Se puede alojar la base de datos en un servidor EC2 dedicado o una instancia con un SGBD en Docker.
    3. Se puede implementar balanceadores de carga de aplicación, tanto con ELB como con una instancia EC2 dedicada
    4. Se puede usar AWS CloudFormation para especificar su infraestructura.
    5. Se puede usar GitHub Actions para implementar un flujo de despliegue continuo
    6. Se puede usar AWS CodeDeploy
    7. Se puede instalar un servidor FTP asegurado mediante FTPS o SFTP
    8. Se puede instalar un WAF como OWASP ModSecurity
    9. Se puede configurar DNS dentro de AWS tanto con Route53 como con un servidor dedicado

4. Diseño Interfaces Web

Los criterios mínimos para el proyecto integrado en el módulo “Desarrollo de Interfaces” son: Crear una Landing Page para el proyecto como presentación del proyecto o descripción de la aplicación con los siguientes requisitos:

DISEÑO

Diseñar un prototipo de una página web con las siguientes características:

Objetivos

1. Crear en FigJam la siguiente estructura:
   - Objetivos del proyecto (Hipótesis, punto de partida e idea).
   - Análisis del público objetivo.
   - Estudio de la competencia.
   - User Flow.
2. Crear un wireframe en Figma de baja fidelidad con las pantallas expuestas en el User Flow.
3. Crear un UI Kit en Atomic Design con la siguientes estructura:
   - Átomos: Tipográfica, Color, Iconos, sistema de rejilla e imágenes.
   - Moléculas.
   - Organismos.
4. Crear un prototipo, diseño en alta fidelidad del wireframe creado en Figma utilizando los componentes del UI Kit .

Documentación

Crear un documento anexo al proyecto donde explique los siguientes apartados.

- Descripción de la temática para vuestra composición. Hipótesis de partida, propósito del sitio web y las metas que se quieren alcanzar.
- Estudio de competencia y sitios de referencia.
- Buscar imágenes acordes a la temática.
- Seleccionar un color dominante asociado a esa temática. Explica la elección de ese color.
- Crea una paleta de colores con el color dominante seleccionado. La paleta podrá ser monocromática, análoga o complementaria. Explicar el uso de los colores y su proporción en una captura de pantalla.
- Selecciona una fuente para el prototipo. Explica la elección de esa fuente. Crea una combinación armónica con otra fuente, aplicando diversos pesos en la jerarquía visual.
- Seleccionar dos colores para la fuente y background, calculando el contraste entre ellos según el estándar WCAG 2.
- Aplicar varios ejemplos en vuestra composición de “Equilibrio visual y tensión compositiva”. Explica la elección de esos elementos y qué significado aporta.
- El prototipo será diseñado para una resolución de 1920 x 1080, utilizando una rejilla de 12 columnas.

- Se aplicarán elementos de accesibilidad y usabilidad en la composición. Razona la elección de dichos elementos.

Especificaciones

- El prototipo tendrá como mínimo una página principal y 4 páginas adyacentes según el User Flow. Las páginas estarán enlazadas con animación.
- Los componentes de la interfaz estarán bien definidos (cabecera, cuerpo, barra lateral, pie de página, espacios en blanco).

CSS3

- Utilizar CSS3 para dar formato a la interfaz. No se podrá utilizar ningún tipo de Framework tipo Tailwind, Bootstrap, Materialize, UIKit, etc . En caso de usar uno de estos frameworks, crear una landing page con los requisitos mínimos expuestos en este documento.
- Crear una animación o transición interactiva de uno o varios elementos del DOM por medio del scroll ó ratón. (Por ejemplo efecto parallax).
- El diseño de la interfaz deberá ser “Responsive” utilizando Media Queries, FlexBox y Grid layout.
- Utilizar el preprocesador SASS para estructurar los archivos css en un único main.css (main.scss) con @import a los demás scss (colores, cabecera, pie, cuerpo…).
- Utilizar la metodología BEM para la descripción de los selectores, variables...

HTML5

- Insertar un elemento multimedia de cada tipo: video, sonido, canvas y SVG.
- Se podrán utilizar las librerías Chartjs, Animejs, D3.js, KoolChart, Snap.svg, Phaser o cualquier otra para incluir gráficas, animaciones, galería de imágenes.

INKSCAPE SVG

- Utilizar un software tipo inkscape para crear el logo vectorial al cual se le podrá aplicar alguna animación con Anime.js para cubrir el punto anterior.

5. Desarrollo Web en Entorno Cliente

Especificación de Requisitos para la Implementación del Entorno Cliente

1. Introducción

El objetivo de este proyecto es desarrollar la parte cliente de una aplicación web utilizando Angular o cualquier otro famework o librería vista en la fase DUAL y de FCT.

Este documento especifica los requisitos mínimos que deben cumplirse y los requisitos adicionales para obtener una calificación superior.

---

2. Requisitos Mínimos

2.1. Estructura y Configuración del Proyecto

- Se debe emplear Git y GitHub para el control de versiones, subiendo código actualizado semanalmente.
- Se usará el sistema de construcción propio .
- Se requiere un README en GitHub con:
    - Título y descripción del proyecto.
    - Objetivos y tecnologías utilizadas.
    - Bitácora de tareas realizadas por cada integrante.
    - Bibliografía y enlaces a documentación de apoyo.

2.2. Gestión de Rutas y Navegación

- Uso del módulo de routing (`RouterModule`) con navegación entre componentes.

2.3. Autenticación y Seguridad

- Implementación de login y registro de usuarios.
- Uso de Token almacenado en LocalStorage o SessionStorage.
- Implementación de un módulo de administración accesible solo por usuarios con rol de administrador.

2.4. Comunicación entre Componentes

- Preferiblemente se debe utilizar Observables para la comunicación entre componentes.
- Dependiendo de la necesidad del componente y la estructura de comunicación, se podrán usar decoradores (`@Input()`, `@Output()`).

2.5. Consumo de API REST

- Uso de HttpClientModule para realizar peticiones a la API.
- Implementación de métodos GET, POST, PUT y DELETE para gestionar datos desde el cliente.
- El consumo de API se debe utilizar para rellenar automáticamente los contenidos de la aplicación, facilitando así su prueba.
- Manejo de errores en las peticiones mediante `try...catch`.

2.6. Almacenamiento en el Navegador

- Uso de LocalStorage y SessionStorage para la gestión del acceso de usuarios con tokens de autenticación.
- Conexión con API implementada en PHP para gestión de una base de datos en json.

---

3. Requisitos Adicionales

Para aquellos alumnos que deseen una mayor calificación y profundizar en el desarrollo moderno con Angular, se establecen los siguientes requisitos adicionales:

3.2. Optimización de Estado y Reactividad

- En clase se han visto Decoradores (`@Input`, `@Output`) y Observables.
    - Se podrá optimizar el código migrando a Señales (Signals) en Angular 17+ si el alumno lo desea.

3.3. Uso de Material Design o Componentes UI Avanzados

- En clase se ha utilizado Bootstrap.
- Se recomienda que los estudiantes investiguen Material Design o PrimeNG para mejorar la interfaz gráfica.

3.4. Persistencia y Seguridad Avanzadas

- En clase se han visto:
    - Cookies, LocalStorage, SessionStorage e IndexedDB.
    - Conexión con API implementada en Flask (Python) para gestión de una base de datos en MySQL.

3.5. Automatización del Despliegue

- No se ha visto en clase y pertenece al módulo de Despliegue de Aplicaciones Web, por lo que no es necesario.