México - Desarrollo Java
Proyecto integrador (60 horas)

Opciones de proyecto

1) **E-Commerce**
2) **Red Social**

Debe haber 6 páginas en cualquiera de las dos opciones de proyecto:

1. Página principal (Inicio) - Página estática con App Presentation
2. Contáctenos - Páginas estáticas
3. Acerca de nosotros - Páginas estáticas
4. Lista de productos / Publicaciones - Consumos API (POST y GET)
5. Login
6. Página de formulario (Ver / Actualizar / Crear / Eliminar)

Tarea # 1

Título: Diseño de Wireframes

Duración: 3 horas.

Secuencia: Después - 2.2.8 - Recapitulación / Revisión

Objetivo: Definir la estructura del proyecto (layouts y navegación)

Descripción:

- Crea los wireframes para el producto que se creará como proyecto final
- Herramienta: https://ninjamock.com

Assessment:

- Contiene formulario de Login
- Contiene formulario de registro
- Contiene formulario de contacto
- Considera los fundamentos de UX y UI

Tarea # 2

Título: Página sobre nosotros

Duración: 3 horas.

Secuencia: Después - 2.2.8 - Recapitulación / Revisión

Objetivo: Implementar la página Acerca de nosotros

Descripción:

- Crea la página Acerca de nosotros usando Bootstrap con:
  - Una breve descripción del proyecto.
  - Perfiles del equipo (breve biografía, foto y papel en el proyecto)
- Herramientas:
  - HTML5
  - Bootstrap

Assessment:

- Utiliza al menos 1 animación o efecto usando CSS / JS.
- Los desarrolladores no dan ninguna advertencia sobre los errores de uso de HTML.
- Considera los fundamentos de UX y UI

Tarea # 3

Título: Página de contáctenos

Duración: 3 horas.

Secuencia: Después - 2.2.8 - Recapitulación / Revisión

Objetivo: Implementar la página Contáctenos

Descripción:

- Crea la página Contáctenos con Bootstrap con los siguientes inputs en el formulario:
    - Nombre
    - Correo electrónico
    - Teléfono
    - Mensaje
- Implementa una función de JavaScript que valide los tipos de entrada y la corrección cuando se presiona el botón Submit
- Implementa una función de JavaScript que envíe los datos de los inputs a un correo electrónico una vez que se pase los inputs de validación
- Herramientas:
    - HTML5
    - Bootstrap
    - JavaScript

Assessment:

- Los desarrolladores no dan ninguna advertencia sobre los errores de uso de HTML.
- Considera Fundamentos de UX y UI

- Funciones de JavaScript de validaciones adecuadas para los form fields dados.

Tarea # 4

Título: Estructura Navegación - Bootstrap + Javascript

Duración: 5 horas.

Secuencia: Después - 3.3.5 - Recapitulación / Revisión

Objetivo: Implementar la estructura básica de navegación de páginas web usando Bootstrap + JavaScript

Descripción:

- Crea la estructura básica para el proyecto con la navegación (crear páginas sin contenido para cada sección):
  - Página Principal
  - Lista de Items
  - Formulario de Creación / Edición del modelo
    - Utilizar Navigation Bar para conectar páginas
- Crea un repositorio para tu código de front-end (HTML + JavaScript + CSS) con los siguientes directorios:
  - /html
  - /css
  - /js
- Conecta la navegación con las páginas creadas en las Tareas 2 y 3 (Acerca de nosotros y contáctenos)
- Prueba tu aplicación web:
  - Navegación por todos las páginas.

    - Páginas agregadas (Acerca de nosotros y contáctenos)
- Da commit y envía tus cambios al repositorio
- Herramientas:
    - Bootstrap
    - Javascript

Assessment:

- La navegación de páginas funciona correctamente.
- Uso del Navigation Bar.
- La aplicación utiliza al menos 3 componentes de Bootstrap

Tarea # 5

Título: Característica 1 - Listado de Objetos del Modelo (Publicación o Producto) - Front-end

Duración: 5 horas.

Secuencia: Después - 3.3.5 - Recapitulación / Revisión

Objetivo: Implementar la función de listado objects del proyecto seleccionado:

- Listado de productos
- Listado de publicaciones

Descripción:

- Comparte el código de referencia para facilitar la carga los ítems de la lista: https://github.com/generation-org/PROJECT/tree/master/front-end-bootstrap
- Utiliza la función *addI*[*https://github.com/generation-org/PROJECT/tree/master/front-end-bootstrap*](https://github.com/generation-org/PROJECT/tree/master/front-end-bootstrap)*tem* del *main.js* para agregar los elementos a la lista con la siguiente estructura json:
    - {'name':'Tayto', 'img':'https://www.irishtimes.com/polopoly_fs/1.4078148!/image/image.jpg', 'description':'Cheese & Onion Chips'}
- Crea 10 objetos JavaScript de muestra y verifica que se agreguen a la lista.
- Da commit y envía tus cambios al repositorio
- Herramientas:
    - Bootstrap
    - Javascript

Assessment:

- La lista de elementos es generada de manera correcta.
- La navegación de la aplicación funciona correctamente
- La Lista de componentes muestra una lista de 10 productos en formato json con el layout definido en los wireframes.

Tarea # 6

Título: Formulario de Creación de objeto del modelo (Publicación o Producto)

Duración: 3 horas.

Secuencia: Después - 3.3.5 - Recapitulación / Revisión

Objetivo: Implementar un formulario para creación del modelo de datos utilizando Bootstrap.

Descripción:

- Crea un formulario con los campos requeridos para la creación del modelo (publicación o producto) utilizando los componentes de formularios de Bootstrap.
- Implementa una función Javascript que valide los campos del formulario y muestre errores en caso de existir usando los Alertas de Bootstrap.
- Una vez validados los campos crea un objeto javascript en formato json con toda la información del formulario.
- Da commit y envía tus cambios al repositorio
- Herramientas:
    - Bootstrap
    - Javascript

Assessment:

- El formulario tiene todos los campos del modelo de datos definidos en los wireframes.
- La función de javascript para validar los datos funciona correctamente.
- Se hace uso de al menos 2 componentes diferentes de formularios de Bootstrap
- Se crea el modelo de datos correcto utilizando la notación json con todos los campos del formulario.

Tarea # 7

Título: Registro de usuarios - Front End

Duración: 4 horas.

Secuencia: Después - 3.3.5 - Recapitulación / Revisión

Objetivo: Implementar la función de registro de usuarios

Descripción:

- Implementa el formulario con Bootstrap de registro de usuario con los siguientes campos
    - Nombre completo
    - Número de teléfono
    - Email (nombre de usuario)
    - Contraseña
- Implementar función Javascript para validaciones de formulario:
    - Contraseña coincide
    - Campos no válidos
    - Validación de correo electrónico
    - Validación del teléfono
    - Mostrar errores usando Alertas de Bootstrap.
- Una vez validado el formulario deberá crear un objeto json con los campos del usuario.
- Da commit y envía tus cambios al repositorio
- Herramientas:
    - Bootstrap
    - Javascript

Assessment:

- El formulario de registro funciona y las validaciones se realizan correctamente

- Los mensajes de error se muestran cuando se envían datos incorrectos usando Alertas de Bootstrap.
- Los campos del formulario son iguales a los de las especificaciones
- Se crea el modelo del usuario usando el formato Json.

Tarea # 8

Título: Inicio de sesión de usuarios - LocalStorage

Duración: 5 horas.

Objetivo: Implementar la función de inicio de sesión del usuario.

Descripción:

- Implemente un formulario de inicio de sesión del usuario
- Realizar las validaciones del formulario:
  - Campos vacíos: nombre de usuario y contraseña
  - Valida los datos enviados con nombre de usuario y contraseña codificados
  - Muestra nombre de usuario y contraseña inválidos
- Almacenar los datos del usuario de prueba en el local storage
- Implementar autenticación verificando usuarios pre almacenados en el local storage.
- Da commit y envía tus cambios al repositorio
- Herramientas:
  - Bootstrap
  - Javascript

- Local Storage API

Assessment:

- Las validaciones del formulario son realizadas cuando se envían datos y se muestran los mensajes de error adecuados.
- Los datos de usuario son verificados usando los datos pre almacenados en el local storage
- Después de iniciar sesión correctamente, redirige a la página de inicio

Tarea # 9

Titulo: Database Structure - MySQL

Duración: 5 horas.

Secuencia: Después - 5.1.4 - Recapitulación / Revisión

Objetivo: Definir e implementar la base de datos del proyecto

Descripción:

- Crea el diagrama de la entidad de la base de datos del proyecto utilizando el workbench MySQL.
- Ejecuta los scripts para crear la base de datos del proyecto.
- Crea los scripts SQL para insertar datos de muestra para cada tabla (5 entradas por tabla)
- Herramientas:
  - MySQL
  - MySQL Workbench

Assessment:

- El diagrama de Entity Relationship se crea con al menos 4 entidades.
- Las declaraciones SQL se pueden ejecutar sin errores.
- El script de creación de base de datos se puede ejecutar sin errores.

Tarea # 10

Título: Proyecto base de Spring Boot API - Backend

Duración: 5 horas.

Secuencia: Después - 5.2.4 - Recapitulación / Revisión

Objetivo: Implementar la estructura básica para el proyecto Spring Boot API.

Descripción:

- Cree un repositorio para tu código de back-end (proyecto Spring Boot)
- Genere un nuevo proyecto Spring Boot utilizando Spring Initializer e incluyendo Web dependency
- Agrega el código al repositorio que creó para el backend
- Implementa el modelo de proyecto POJOS (Usuarios, Producto o Publicación)
- Implemente UserController y UserService con los siguientes endpoints:
    - Login
    - Registro
- Implemente el ModelController (PostController o ProductController) con los siguientes puntos finales:

    - Crea el nuevo objeto modelo (POST)
    - Lee el objeto modelo (GET)
    - Actualiza el objeto modelo (PUT)
    - Elimina el objeto modelo (BORRAR)
    - Lee todos los objetos modelo (GET)
- Da commit y envía tus cambios al repositorio
- Herramientas:
    - Spring Initializer
    - Spring Framework
    - IntelliJ Idea Community Edition

Assessment:

- La clase Users Controller se crea y no proporciona ningún error de compilación
- El Model Controller (productos o publicaciones) se crea y no da ningún error de compilación.
- La aplicación Spring Boot compila y se ejecuta con `mvn spring-boot:run`
- Spring Boot Controllers se crean utilizando @Controller

Tarea # 11

Título: Spring Data JPA - Backend

Duración: 4 horas.

Secuencia: Después - 5.2.4 - Recapitulación / Revisión

Objetivo: Conectar la base de datos MySQL creada en la Tarea 9 con el proyecto Spring Boot

Descripción:

- Implementa UserRepository que se conecta con la base de datos del usuario.
- Implementar ModelRepository (Publicación o Producto) que se conecta con la base de datos del modelo de usuario (publicación o producto)
- Implementa los métodos requeridos para cada repositorio de las operaciones CRUD en la base de datos requerida para los controladores en la tarea 11
- Da commit y envía tus cambios al repositorio

- Herramientas:
  - Spring Initializer
  - Spring Framework
  - IntelliJ Idea Community Edition

Assessment:

- La conexión con la base de datos funciona y permite la manipulación del Spring Boot Project
- La aplicación Spring Boot compila y se ejecuta con `mvn spring-boot:run`
- La clase ModelRespository puede acceder a la tabla de modelos y realizar operaciones CRUD.

Tarea # 12

Título: Spring Boot +Front End + Deployment

Duración: 3 horas.

Secuencia: Después - 6.1.6 - Recapitulación / Revisión

Objetivo: Conectar back-end con front-end

Descripción:

- Despliega el proyecto de Spring Boot como aprendiste en el módulo de Deployment
- Instala la siguiente extension de Firefox y trabaja en este navegador para evitar la validación de seguridad e CORS orign: https://addons.mozilla.org/en-US/firefox/addon/cors-everywhere/
- Implementa la llamada del API desde Javascript usando fetch del endpoint de registro de usuario
- Implementa la llamada del API desde Javascript usando fetch del endpoint de inicio de sesión del usuario
- Implementa la llamada del API desde Javascript usando fetch de la lista de objetos modelo
- Implementa la llamada del API desde Javascript usando fetch desde el formulario de creación del modelo(productos o publicaciones)
- Da commit y envía tus cambios al repositorio
- Herramientas:
    - Heroku
    - Spring Framework
    - IntelliJ Idea Community Edition
    - Fetch- Javascript

Assessment:

- Se proporciona una url para acceder al proyecto y verificar que funciona.
- Los componentes de inicio de sesión validan las credenciales del usuario y verifican los datos de los usuarios en el back-end.
- Se devuelven los datos del usuario de las bases de datos.

Tarea # 13

Título: Unit Tests

Duración: 4 horas.

Secuencia: Después - 6.2.6 - Recapitulación / Revisión

Objetivo: Implementar la Unit test para REST API

Descripción:

- Implementar Unit Tests para los siguientes componentes:
  - ModelController (ProductsController o PostsController)
  - ModelService (ProductsService o PostsService)
- Ejecuta tu Unit Test y presenta tus resultados.
- Da commit y envía tus cambios al repositorio

- Herramientas:
  - JUnit5
  - Spring Framework
  - IntelliJ Idea Community Edition

Assessment:

- Crear 5 entradas de datos de muestra para llenar cada tabla de su aplicación e implementar pruebas unitarias para sus repositorios con estos datos.
- Todas las pruebas deben pasar (color verde)
- Todos los endpoints se prueban con al menos 2 unit tests

Tarea # 14

Título: Presentaciones finales del proyecto

Duración: 1 día completo

Secuencia: después de completar todas las tareas, idealmente el último día de la semana 12

Objetivo: Presentar proyectos a sus pares y otras partes interesadas relevantes.