

Proyecto Integrador

Objetivo

Cada equipo deberá definir la temática que ofrecerá el E-commerce que se desarrollará en el proyecto integrador.

Al finalizar el Proyecto Integrador el mismo contará con los siguientes módulos:

El FrontEnd deberá proveer:

- Home del sitio.
- Página con formulario de login.
- Página con formulario de registración donde se registran los datos de los usuarios como así también se definirá qué roles tendrá para usar la aplicación.
- Perfil de usuario: Mail, contraseña, foto. (Otros, sin límite).
- Preguntas frecuentes.
- Formulario de contacto.

- Carrito de compra: página del carrito, funcionalidad para eliminar un producto del carrito, cerrar la compra (cotización).
- Validaciones importantes: evitar que el dueño del producto pueda auto comprar su producto.

El Backend deberá:

- Proveer una API para comunicarse con el FrontEnd.

- Manejo de excepciones.
- Logueo de errores.
- Control de seguridad según los roles de los usuarios.

Módulo integrado:

- Panel de Administrador

Además se deberá desarrollar un módulo con thymeleaf para la parte visual y para el backend todo lo visto de java donde podamos realizar un ABM de Productos: título, descripción y foto (1 o más), precio

Para la aprobación del PI los equipos deberán cumplir con:

- Las entregas pautadas por las fechas de fin de sprint.
- Cada equipo creará un repositorio GIT el cual será el medio para la entrega.
- Utilizar en el desarrollo del proyecto las metodologías ágiles. (presentando en todo momento al docente un trello con la planificación del mismo)
- Además para el desarrollo del FrontEnd se verificará que se utilice un diseño responsive centrado en la metodología mobile first.
- Cada módulo debe poder navegarse entre sí de manera fluida.
- Cada equipo deberá presentar un Panel de Administración.

Metodología para el desarrollo de los proyectos integradores:

En el marco de simular el ambiente laboral trabajaremos con metodologías ágiles, especialmente con scrum. Aplicando las metodologías ágiles vamos a dividir las entregas en Sprint los cuales van a estar fijados con una duración de 15 días.

Cada Sprint tendrá una duración de dos semanas, en donde se les dará a cada grupo consignas obligatorias a entregar. Se debe dejar asentado en la planilla de evaluación dichos eventos durante la cursada.

Durante cada Sprint, en forma optativa, se puede llevar a cabo el siguiente momento:

3

Daily meetings Cada día, los/as alumnos/as destinan 15 minutos (máximo) para sincronizar el trabajo diario. La idea es que lo hagan por fuera del espacio de clase. Cada integrante del equipo responde a 3 preguntas simples:

¿Qué hice ayer?

¿Qué haré hoy?

¿Qué impedimentos o dificultades tengo?

El objetivo no es que nos reporten sus progresos. La daily meeting es una reunión exclusiva del equipo de desarrollo y se trata de que, como equipo, puedan identificar qué problemas se han encontrado. Es responsabilidad del scrum master de asegurarse que se genere esta reunión. Aquí es, por ejemplo, cuando podrán identificar si algún miembro del grupo está teniendo dificultades. Como profes deberán transmitir la importancia de esta instancia para el trabajo con metodologías ágiles y brindarles dinámicas para la realización de las dailys, en caso de que sea necesario.

Recomendamos que utilicen el siguiente tablero:

BackLog	Sprint	Doing	Ready to test	Done

Planificación de los sprint

Sprint 1

Objetivo

Realizar el FrontEnd de nuestra aplicación.

Cada equipo deberá definir la temática de su proyecto optando por Ecommerce.

Como dueños del producto, necesitamos que nuestro sitio web cuente con la posibilidad de mostrar en la sección de inicio un diseño lindo y estético que hable acerca de lo que hace nuestra plataforma digital. (Home Page)

De igual manera queremos contar con una sección en la cual nuestros usuarios encuentren respuestas a las preguntas frecuentes que solemos recibir acerca de nuestros servicios ofrecidos. (FAQ)

También necesitamos que nuestra página web tenga dos formularios que permita que los usuarios por un lado, puedan registrarse a nuestra plataforma y por otro loguearse a la misma. (Login, Registración de usuario)

Finalmente queremos ver también una sección que nos muestre como un usuario vería su perfil cuando ingrese a nuestro sistema.(Perfil de usuario)

Entregables del sprint

- Web responsive de todas las secciones - Visualmente estético

Obligatorias para el 1er Sprint

Home

F.A.Q. - Preguntas frecuentes

Formulario de Registro

Formulario de Login

Formulario de Contacto

Perfil de Usuario

Una sección en donde se listen todos los productos de nuestro sistema.

Una sección en donde se pueda ver el detalle de un producto.

Opcionales:

Una sección en donde podamos ver cómo sería el carrito de compra de nuestra web.

- Cada equipo creará un **repositorio GIT** el cual será el medio para la entrega

Para corregirlo y darte feedback, tus profes tendrán en cuenta los siguientes criterios:

- Que la estructura HTML esté definida utilizando tags semánticos
- En cuanto al estilo CSS que la definición de los mismos esté optimizada y sin repetir reglas.
- Que contenga: formulario de registro, login y contacto
- Que haya una sección de preguntas frecuentes
- Que cuente con un perfil de usuario
- Navegación fluída
- El sitio tiene que ser responsive
- Tiene que contar con todas las vistas requeridas para el tipo de sitio que están realizando
- Por último, no se olviden de usar GIT!

Sprint 2

Objetivo

Para este spring se debe incorporar en sus respectivo desarrollos aquellos requerimientos que no fueron alcanzados en el sprint I.

Planificar el consumo por medio de un un archivo Json, de los diferentes endponit para obtener un sitio dinámico.

Web responsive de varias secciones - Visualmente estético II

Home

F.A.Q. - Preguntas frecuentes

Formulario de Registro

Formulario de Login

Formulario de Contacto

Perfil de Usuario

Vista de listado de productos

Vista del detalle de un producto

Vista del carrito de compra

Sprint 3

Objetivo

Para este spring se debe incorporar en sus respectivo desarrollos aquellos requerimientos que no fueron alcanzados en el sprint 2.

La utilización de bases de datos relacionales SQL aporta muchos beneficios. Seguridad, performance y escalabilidad entre varios otros. El objetivo principal del sprint es definir e implementar el DER para nuestra aplicación en desarrollo.

Requerimientos obligatorios

Para todos los modelos de negocio:

1. Diseñar e implementar un DER con el diseño de la base de datos de nuestra aplicación.
 - A/B/M de productos.

Sprint 4

Objetivo

Para este spring se debe incorporar en sus respectivo desarrollos aquellos requerimientos que no fueron alcanzados en el sprint 3.

Para esto sprint debemos realizar un diagrama UML de clases de toda nuestra aplicación y luego implementarlo en java.

Dado que este sprint es crítico para el funcionamiento de nuestra aplicación, ya que estamos codificando el backend de la misma, debemos realizarlo en una nuevo repositorio GIT.

Requerimientos obligatorios

- Abrir un nuevo branch en nuestro repositorio GIT.
- Realización de diagrama de clases UML de la aplicación.
- Utilizar herencia, polimorfismo e interfaces de necesitarlas.
- Se deben aclarar:
 - a. Clases abstractas.
 - b. Interfaces.
 - c. Relaciones de uso y de tiene.
 - d. Tipo de dato que reciben y devuelven los métodos.
 - e. Tipos de datos de los atributos.
 - f. Visibilidad de métodos y atributos.(público, privados,protegidos).
 - g. Herencias.
 - h. Polimorfismo.
- Implementar completamente en java el diagrama de clases definido en el spring.

Requerimientos opcionales

- Si mi aplicación tendrá distintos tipos de usuarios (administrador, usuario, etc). Puedo plasmarlo en el diagrama de clases?
- Es aplicable algún tipo de patrón de diseño a mi aplicación? De ser así intenta diagramarlo.
- En caso de ser necesarias validaciones deberán desarrollarlas.

Sprint 5

Objetivo

Para este spring se debe incorporar en sus respectivo desarrollos aquellos requerimientos que no fueron alcanzados en el sprint anterior.

Sabemos que la utilización de un Framework como **Spring Boot**, y la implementación de un patrón de arquitectura MVC nos beneficia en muchos sentidos, ordenando nuestra aplicación, dándole escalabilidad y acelerando el desarrollo de nuevas features entre otras cosas.

Nuestro objetivo en éste sprint será la migración de nuestro proyecto anterior utilizando spring boot. Este sprint representa un cambio radical en nuestra aplicación, podríamos considerarlo una nueva versión de nuestra aplicación. Por esto, decidimos abrir un nuevo branch con la nueva versión de nuestra aplicación donde desplegaremos spring boot. utilizando Rest, y desarrollando los diferentes endpoint que nos permitirán comunicarnos con el frontend por tal motivo debemos:

Requerimientos obligatorios

- Abrir un nuevo branch en nuestro repositorio GIT.
- Implementar Spring Boot junto con los servicios rest que nos permitan interactuar con el front.
- **Lógica:** Generación de controladores. Creación de objetos y métodos para su uso.
- **Base de datos:** Implementación de modelos, usar Spring Data.
- **Vista:** Utilizando thymeleaf crear las diferentes vistas para el panel de administración (Según el modelo de negocio).

Sprint 6

Objetivo

Para este spring se debe incorporar en sus respectivo desarrollos aquellos requerimientos que no fueron alcanzados en el sprint anterior.

Sabemos que la utilización de un Framework como **Spring Boot**, y la implementación de un patrón de arquitectura MVC nos beneficia en muchos sentidos, ordenando nuestra aplicación, dándole escalabilidad y acelerando el desarrollo de nuevas features entre otras cosas.

Nuestro objetivo en éste sprint será completar con el desarrollo de las respectivas aplicaciones.

Herramientas sugeridas para llevar adelante nuestro PI:

Planificacion y coordinacion de tareas:

Trello: <https://trello.com>

Mokear API:

My JSON Server: <https://my-json-server.typicode.com/>