

PROYECTO FINAL INTEGRACIÓN DE APLICACIONES

Tienda de Mascotas: Uso de los conceptos de Integración de Aplicaciones

En el dinámico mundo de la tecnología moderna, la integración efectiva de aplicaciones se ha convertido en una habilidad fundamental para los desarrolladores de software. Este proyecto de implementación de una tienda de mascotas online ofrece una oportunidad única para aplicar y sintetizar los conceptos clave de la integración de aplicaciones en un escenario del mundo real.

Los estudiantes se enfrentarán al desafío de diseñar e implementar una plataforma de comercio electrónico completa, abarcando desde la gestión de productos y usuarios hasta el procesamiento de pedidos y sistemas de recomendación. A través de la creación de una API RESTful robusta, la integración de múltiples servicios, y la implementación de flujos de datos complejos, los participantes ganarán experiencia práctica en arquitecturas de microservicios, seguridad de aplicaciones, y técnicas de optimización de rendimiento.

Este proyecto no sólo consolida los fundamentos teóricos de la integración de aplicaciones, sino que también expone a los estudiantes a las mejores prácticas de la industria y a tecnologías de vanguardia. Al completar este proyecto, los estudiantes estarán bien equipados con habilidades altamente demandadas en el mercado laboral actual, preparándolos para enfrentar los desafíos del desarrollo de software moderno y la integración de sistemas complejos.

INTEGRACIÓN DE APLICACIONES

PROYECTO FINAL INTEGRACIÓN DE APLICACIONES..... 1

Tienda de Mascotas: Uso de los conceptos de Integración de Aplicaciones..... 1

1. PUNTOS CLAVES DE APRENDIZAJE..... 3

Aplicación Práctica de Conceptos Teóricos.....3

Experiencia con Arquitecturas Modernas..... 3

Integración de Múltiples Servicios..... 3

Manejo de Datos en Tiempo Real..... 3

Seguridad y Autenticación..... 3

Escalabilidad y Rendimiento.....3

Manejo de Datos Complejos.....3

Desarrollo de API.....3

Relevancia en el Mercado Laboral..... 3

Desafíos del Mundo Real..... 4

Base para Futuros Proyectos..... 4

Conclusión..... 4

2. MATERIALES PROPORCIONADOS.....4

• <https://github.com/jlfg-evereven/ucjc-ida/tree/main/petstore-final-project/data>..... 4

3. IMPLEMENTACIÓN.....6

Requerimientos de Implementación..... 6

1. PUNTOS CLAVES DE APRENDIZAJE

Aplicación Práctica de Conceptos Teóricos

Este proyecto os ofrece una excelente oportunidad para aplicar los conceptos teóricos de integración de aplicaciones en un escenario del mundo real. Podréis ver cómo los principios aprendidos en clase se traducen en soluciones prácticas y funcionales.

Experiencia con Arquitecturas Modernas

La implementación de una tienda en línea de mascotas os permitirá trabajar con arquitecturas modernas como microservicios y APIs RESTful. Estas son habilidades altamente demandadas en la industria actual del desarrollo de software.

Integración de Múltiples Servicios

El proyecto involucra la integración de diversos servicios (productos, usuarios, carrito de compras, pedidos, etc.), lo que proporciona una comprensión profunda de cómo diferentes componentes de un sistema complejo interactúan entre sí.

Manejo de Datos en Tiempo Real

La implementación de funcionalidades como la gestión del carrito de compras y el procesamiento de pedidos os puede ofrecer la oportunidad de trabajar con actualizaciones de datos en tiempo real, un aspecto crucial en las aplicaciones modernas.

Seguridad y Autenticación

Al manejar información de usuarios y procesar pedidos, aprenderéis sobre la importancia de la seguridad en la integración de aplicaciones, incluyendo autenticación y autorización.

Escalabilidad y Rendimiento

El diseño de un sistema capaz de manejar múltiples usuarios, productos y pedidos simultáneamente introduce consideraciones importantes sobre escalabilidad (resiliencia) y optimización del rendimiento.

Manejo de Datos Complejos

La variedad de estructuras de datos (productos, pedidos, reseñas, etc.) ofrece experiencia en el manejo y transformación de datos complejos entre diferentes partes del sistema.

Desarrollo de API

Ganaréis experiencia práctica en el diseño e implementación de APIs, una habilidad fundamental en el desarrollo de software moderno.

Relevancia en el Mercado Laboral

INTEGRACIÓN DE APLICACIONES

Las habilidades adquiridas en este proyecto son directamente aplicables a muchos roles en el desarrollo de software, aumentando vuestra empleabilidad.

Desafíos del Mundo Real

Al trabajar en un proyecto que simula una aplicación del mundo real, enfrentaréis desafíos similares a los que os encontraréis en vuestras futuras carreras profesionales.

Base para Futuros Proyectos

Los conocimientos y habilidades adquiridos en este proyecto servirán como una sólida base para futuros proyectos más complejos y especializados en integración de aplicaciones.

Conclusión

Este proyecto de tienda de mascotas no es solo un ejercicio académico, sino una experiencia que os preparará para los desafíos del desarrollo de software en el mundo real. Os proporcionará un conjunto completo de habilidades en integración de aplicaciones que son altamente relevantes y demandadas en la industria tecnológica actual.

2. MATERIALES PROPORCIONADOS

Se adjunta en el siguiente repositorio un conjunto de posibles estructuras de datos que podéis utilizar y datos para poblarlas:

- <https://github.com/jlfg-evereven/ucjc-ida/tree/main/petstore-final-project/data>

Entidades/servicios a implementar:

- Búsqueda
- Productos
- Categorías
- Usuarios
- Carrito
- Pedidos
- Reseñas
- Mascotas (Opcional)

Flujos de comunicación:

```
sequenceDiagram
    participant Cliente
    participant Búsqueda
    participant Productos
    participant Categorías
    participant Usuarios
    participant Carrito
    participant Pedidos
    participant Reseñas
    participant Mascotas
```

INTEGRACIÓN DE APLICACIONES

(1:1) Autenticación del Usuario (Usuarios)

- El cliente inicia sesión y recibe un token de autenticación.

Cliente->>Usuarios: POST /api/users/login
Usuarios-->>Cliente: Token de autenticación

(1:N) Búsqueda de Producto (Búsqueda, Productos, Categorías)

- El usuario busca "collar para perro".
- El servicio de Búsqueda consulta a Productos y Categorías para obtener resultados relevantes.

Cliente->>Búsqueda: GET /api/search?q=collar para perro
Búsqueda->>Productos: Consulta productos
Búsqueda->>Categorías: Consulta categorías
Búsqueda-->>Cliente: Resultados de búsqueda

(1:1) Detalles del Producto (Productos, Reseñas)

- El usuario selecciona un producto específico.
- El servicio de Productos obtiene los detalles y un resumen de las reseñas.

Cliente->>Productos: GET /api/products/12345
Productos->>Reseñas: Obtiene resumen de reseñas
Productos-->>Cliente: Detalles del producto

(1:1) Reseñas del Producto (Reseñas)

- El usuario solicita ver las reseñas completas del producto.

Cliente->>Reseñas: GET /api/products/12345/reviews
Reseñas-->>Cliente: Lista de reseñas

(1:1) Añadir al Carrito (Carrito, Productos)

- El usuario añade el producto al carrito.
- El servicio de Carrito verifica la disponibilidad con el servicio de Productos.

Cliente->>Carrito: POST /api/cart/items
Carrito->>Productos: Verifica disponibilidad
Carrito-->>Cliente: Confirmación de añadido al carrito

(1:1) Ver Carrito (Carrito)

- El usuario revisa el contenido de su carrito.

Cliente->>Carrito: GET /api/cart
Carrito-->>Cliente: Contenido del carrito

(1:N) Realizar Pedido (Pedidos, Usuarios, Carrito, Productos)

- El usuario realiza un pedido.
- El servicio de Pedidos verifica la dirección con Usuarios, obtiene los ítems del Carrito y actualiza el inventario en Productos.

INTEGRACIÓN DE APLICACIONES

Cliente->>Pedidos: POST /api/orders
Pedidos->>Usuarios: Verifica dirección de envío
Pedidos->>Carrito: Obtiene items del carrito
Pedidos->>Productos: Actualiza inventario
Pedidos-->>Cliente: Confirmación de pedido

Recomendaciones Personalizadas (Mascotas, Productos, Categorías) (OPCIONAL)

- El sistema obtiene la lista de mascotas del usuario.
- Basándose en esta información, se buscan productos relevantes (por ejemplo, alimento para perro).

Cliente->>Mascotas: GET /api/pets
Mascotas-->>Cliente: Lista de mascotas del usuario

Cliente->>Productos: GET /api/products?category=alimento&animalType=perro
Productos->>Categorías: Verifica categoría
Productos-->>Cliente: Lista de alimentos para perro

3. IMPLEMENTACIÓN

Requerimientos de Implementación

- (DOCUMENTACIÓN) Utilizar Swagger/OpenAPI para documentar la API (LAB 2) y que os sirva como guía de implementación de los métodos.
- **No hace falta implementar un front** pero tendréis que tener algún mecanismo para probar los flujos solicitados.
- El proyecto se entregará en un repositorio de github donde me proporcionaréis vuestro usuario.

Puntuación por secciones

- (DESARROLLO DE SOFTWARE) Desarrollar los servicios expuestos de manera individual con la tecnología que uno considere y sus APIs (**1p**)
- (COMUNICACIÓN) Entre los servicios expuestos se solicita una serie de flujos de comunicación requeridos, entre los cuales habrá que hacer implementaciones de servicio a servicio (1:1 One to One) y de servicio a varios (1:N One to Many) (**+3p**)
- (ARQUITECTURA) Se valora el uso de algún middleware (**+1p**)
- (DOCUMENTACIÓN) Implementación de una memoria o README.md con scripts para indicar los pasos para hacer funcionar la plataforma y la ejecución de los flujos solicitados (**+1p**)
- Creación del diagrama de arquitectura con drawio con todos los componentes solicitados y sus comunicaciones y adjuntarlo a la raíz del repositorio (**+0,5p**)
- (ARQUITECTURA) Los servicios se ejecutan dentro de contenedores (**+0,5p**)
- (RESILIENCIA) Uso de balanceadores de carga para obtener resiliencia de los servicios e implementación con al menos dos servicios de cada categoría levantados (**+1p**)
- (RESILIENCIA) Gestión de Errores: Implementar manejo de errores consistente en toda la API (**0,5p**)
- (RESILIENCIA) Exposición de todos los servicios a través de un API Gateway (**+0,5p**)
- (SEGURIDAD) Autenticación: Utilizar JWT (JSON Web Tokens) para la autenticación de usuarios (**0,5p**)

INTEGRACIÓN DE APLICACIONES

- (SEGURIDAD) Autorización (RBAC): Implementar roles (por ejemplo, 'usuario' y 'admin') para controlar el acceso a ciertas rutas **(0,5p)**

Extras

- Versionado: Considerar el versionado de la API (por ejemplo, `/v1/api/...`) para futuras actualizaciones.
- Rate Limiting: Implementar límites de tasa para prevenir abusos.

Mejoras de software

- Validación: Implementar validación de datos en todas las rutas POST y PUT.
- Paginación: Implementar paginación en todas las rutas que devuelven listas de elementos.
- Filtrado y Ordenación: Permitir filtrado y ordenación en rutas relevantes (por ejemplo, listado de productos)