|  |  |
| --- | --- |
|  |  |
| OutfitForYou |  |
|  |  |
|  | Abril 19 2024Diseño Detallado de Software |
|  | Vivian Echeverri LondoñoHector Manuel Vanegas |

# Introducción

La intención de este documento es presentar, de manera concisa pero completa, las metodologías y enfoques que guiarán el desarrollo de los componentes, así como los patrones más relevantes a utilizar. Se delimitarán el alcance y contexto del proyecto, estableciendo definiciones clave para el diseño técnico, drivers, requerimientos funcionales significativos, restricciones y atributos de calidad. Además, se explorarán las vistas que conforman la arquitectura, junto con su modelo subyacente, para finalmente abordar los riesgos inherentes a esta estructura.

Desde una perspectiva más amplia este proyecto surge con la idea de abordar diversas problemáticas y necesidades presentes en la sociedad con relación al ámbito de la moda y la vestimenta. La moda, entendida como el “gusto colectivo y cambiante en lo relativo a prendas de vestir y complementos”, ha sido constante a lo largo de la historia humana, siendo mucho más que una mera tendencia pasajera. Por el contrario, ha sido una manifestación de libertad, un requerimiento social, y una expresión tangible de la cultura.

## Objetivo

El objetivo de este documento es establecer el diseño técnico para el proyecto de desarrollo de una aplicación destinada a la creación de outfits personalizados. Este diseño se fundamentará en los dominios de negocio, aplicación. Se buscará proporcionar una estructura sólida y coherente que permita la integración eficiente de los diferentes aspectos del proyecto, desde los requerimientos del negocio hasta las consideraciones tecnológicas más avanzadas. En última instancia, el objetivo es garantizar que la aplicación resultante cumple con los estándares de calidad, seguridad y usabilidad requeridos, ofreciendo una experiencia óptima para los usuarios finales.

## Definiciones, Acrónimos y Abreviaturas

App: Abreviatura de “aplicación”, se refiere a un programa infromativo diseñado para cumplir una función específica en dispositivos móviles u ordenadores.

Outfit: Término en inglés que se refiere a un conjunto de prendas de vestir y accesorios combinados para crear un conjunto armonioso.

Diseño Técnico: Proceso de definición detallada de la estructura, funcionamiento y aspecto visual de una aplicación o sistema informático.

Arquitectura basada en Microservicios: Es un enfoque arquitectónico para desarrollar una aplicación como un conjunto de servicios pequeños e independientes, cada uno ejecutando un proceso único y específico.

Java: Es un lenguaje de programación interpertado, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Spring Boot: Proporciona un conjunto de herramientas para construir rápidamente aplicaciones, basándose en una infraestructura ligera que elimina la mayor parte del trabajo de configurar las aplicaciones basadas en Spring.

Requerimientos Funcional: Define el comportamiento interno del Software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades que definen como los casos de uso serán satisfechos.

Requerimientos No Funcional: Un requerimiento que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos.

Requerimiento Funcional Significativos: Describen las acciones que el sistema debe realizar, las operaciones que debe admitir o las caracteristicas que debe tener para satisfacer las necesidades de los usuarios finales. Implica que estos requerimientos son críticos o importantes para el funcionamiento y la utilidad del sistema. Pueden ser indentificados mediante la colaboración entre los stakeholders del proyecto, como los usuarios, los clientes, los analistas de negocio y los desarrolladores, quienes determinan qué funciones son esenciales para alcanzar los objetivos del proyecto.

## Alcance

El alcance de este proyecto consiste en el desarrollo de una aplicación móvil para la creación de outfits personalizados, con el objetivo de proporcionar a los usuarios una experiencia interactiva y personalizado en el ámbito de la moda y la vestimenta. La aplicación permitirá a los usuarios explorar, seleccionar y personalizar conjuntos de ropa y accesorios de acuerdo con sus preferencias de estilo y tendencias de moda actuales. El alcance del proyecto incluirá las siguientes funcionalidades principales:

* Funcionalidades Principales:

Creación de perfiles de usuario.

Búsqueda y navegación de prendas de vestir y accesorios.

Visualización de conjuntos de ropa recomendados.

Personalización de conjuntos según preferencias de estilo y ocasiones específicas.

Integración con redes sociales para compartir outfits y recibir feedback.

Funcionalidades de compra, incluyendo carrito de compras y proceso de pago.

* Gestión de Contenido:

Base de datos de prendas de vestir y accesorios, incluyendo imágenes, descripciones, precios y tallas disponibles.

Herramientas con servicios de administración de contenido para agregar, editar y eliminar productos.

* Integraciones Externas:

Integración con APIs de proveedores externos de moda para obtener información actualizada sobre productos, precios y disponibilidad de inventario.

Integración con servicios de pago en línea para facilitar transacciones seguras.

Integración con redes sociales para compartir outfits y acceder a datos de perfil de usuario.

* Experiencia del Usuario:

Interfaz de usuario intuitiva y atractiva.

Experiencia de navegación fluida y receptiva.

Personalización de la experiencia según las preferencias del usuario.

Funcionalidades de retroalimentación y soporte al cliente.

Decisión Arquitectura:

La arquitectura de la aplicación se basará en un enfoque de microservicios, que proporcionará una estructura técnica flexible y escalable para el desarrollo y la implementación del sistema. La arquitectura constará de los siguientes componentes principales:

* Fronted

Interfaz de usuario desarrollada utilizando tecnología web como HTML, CSS y JavaScript, posiblemente con un marco de trabajo como React.js o Angular.js.

Lógica del cliente para manejar la interacción del usuario, las solicitudes de servidor y la presentación de datos.

* Backend

Servidor de aplicaciones desarrollado utilizando tecnologías como Java Spring

Base de datos para almacenar información sobre usuarios, productos y transacciones

* Servicios de integración

API RESTful para facilitar la comunicación entre el front y el back, así como la integración con servicios externos.

Integración con APIs de proveedores externos de moda y servicios de pago en línea.

* Motor de recomendaciones

Utiliza algoritmos de recomendaciones para sugerir conjuntos personalizados basados en las preferencias del usuario y las tendencias de moda actuales.

* Seguridad

Implementación de medidas de seguridad como autenticación de usuarios, cifrado de datos y protección contra ataques.

* Infraestructura

Despliegue en la nube utilizando servicios de proveedores como AWS, Google Cloud y Microsoft Azure.

Contenedores Docker para facilitar el despliegue y la gestión en entornos de desarrollo y producción.

La arquitectura propuesta proporcionará una base sólida y escalable para el desarrollo de la aplicación de creación de outfits personalizados, garantizando una experiencia de usuario óptima y cumpliendo con los objetivos del proyecto.

# Contexto

## Fundamentos de la Solución

Los patrones de diseño sirven para dar soluciones comunes y eficientes a problemas de la arquitectura. Los patrones que se utilizan en esta arquitectura son los siguientes:

|  |  |
| --- | --- |
|  |  |
| Forma  Descripción generada automáticamente |  |

Ilustración 1. Patrones de Arquitectura

## Decisiones de Arquitectura

Se proponen los siguientes patrones que se utilizan en la etapa de Arquitectura:

|  |  |  |
| --- | --- | --- |
| Cod.  Decición | Nombre | Descripción |
| DA01 | Microservicios | Se implementa como patrón nuclear el estilo de arquitectura Microservicios, la cual se centra en componentes desacoplados y cada uno con su propia responsabilidad y sus recursos específicos, permitiendo realizar, cuando se requiera escalabilidad de recursos y generando una arquitectura modular permitirá modificar o agregar nuevas funcionalidades al sistema. |
| DA02 | Clean  Arquitecture | Se implementa como patrón complementario Clean Architecture (Hexagonal – Puertos y Adaptadores), con el fin de evitar el acoplamiento de nuestro dominio con elementos externos lo que producirá sistemas:  Independientes del framework y proveedor.  Testables.  Independientes de la UI  Independientes de la base de datos.  Independiente de agentes externos  Aumentar la tolerancia al cambio.  Reutilizables.  Mantenibles. |

## Decisiones de Diseño

Se proponen los siguientes principios que se utilizarán en la etapa de Diseño

|  |  |  |
| --- | --- | --- |
| Cod.  Decisión | Nombre | Descripción |
| DD01 | SRP | Single Responsability Principle o Principio de Responsabilidad Única |
| DD02 | OCP | Open/Closed Principle o Principio de Abierto/Cerrado |
| DD03 | LSP | Liskov Substitution Principle o Principio de Sustitución de Liskov |
| DD04 | ISP | Interface Segregation Principle o principio de Segregación de la Interfaz |
| DD05 | DIP | Dependency Inversión Principle o Principio de Inversión de Dependecia |

# Drivers de Arquitectura

## Objetivos de la Arquitectura de la solución

Los objetivos de la arquitectura de la solución son las metas y principios que guían el diseño y la implementación del sistema. Estos objetivos se centran en garantizar que la arquitectura satisfaga las necesidades del proyecto y cumpla con los requisitos de los stakeholders. Algunos de los objetivos comunes de la arquitectura de la solución para una aplicación de creación de outfits personalizados podrían incluir:

* Escalabilidad: La arquitectura debe ser capaz de crecer y adaptarse a medida que aumenta la cantidad de usuarios y la carga de trabajo, garantizando un rendimiento óptimo en todo momento.
* Flexibilidad: Debe ser flexible y modular, permitiendo la incorporación de nuevas funcionalidades y la integración con sistemas externos de manera sencilla y sin afectar a otras partes del sistema.
* Seguridad: Debe garantizar la seguridad de los datos del usuario, protegiendo la información personal y financiera contra accesos no autorizados y ataques maliciosos.
* Usabilidad: La arquitectura debe proporcionar una experiencia de usuario intuitiva y agradable, asegurando que la aplicación sea fácil de usar y accesible para usuarios de diferentes niveles de habilidad.
* Mantenibilidad: Debe ser fácil de mantener y actualizar, permitiendo la identificación y corrección rápida de errores, así como la implementación de mejoras y nuevas características de manera eficiente.
* Rendimiento: Debe garantizar un rendimiento rápido y eficiente, minimizando los tiempos de carga y respuesta para ofrecer una experiencia fluida y receptiva a los usuarios.
* Integración: Debe facilitar la integración con sistemas externos, como APIs de proveedores de moda y servicios de pago en línea, permitiendo una colaboración fluida y una experiencia de usuario mejorada.

Al definir y cumplir con estos objetivos, la arquitectura de la solución asegura que la aplicación de creación de outfits personalizados sea robusta, segura y altamente funcional, cumpliendo con las expectativas de los usuarios y los requisitos del negocio.

## Atributos de Calidad

### Extensibilidad

|  |  |
| --- | --- |
| *Extensibilidad* | |
| *Definición* | *Facilidad de adaptar el producto software a los cambios de especificación.* |

### Reusabilidad

|  |  |
| --- | --- |
| *Reusabilidad* | |
| *Definición* | *Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.* |

### Modularidad

|  |  |
| --- | --- |
| *Modularidad* | |
| *Definición* | *Facilidad de adaptar el producto software a los cambios de especificación.* |

### Interoperativad

|  |  |
| --- | --- |
| *Interoperatividad* | |
| *Definición* | *Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.* |

### Escalabilidad

|  |  |
| --- | --- |
| *Extensibilidad* | |
| *Definición* | *Capacidad de un sistema para funcionar bien cuando se presentan cambios en la demanda en la carga de este. Típicamente el sistema será capaz de extenderse a un número mayor o más poderosos de servidores al incrementarse la demanda o la carga.* |

### Usabilidad

|  |  |
| --- | --- |
| *Usabilidad* | |
| *Definición* | *Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.* |

### Rendimiento

|  |  |
| --- | --- |
| *Rendimiento* | |
| *Definición* | *Capacidad del sistema de hacer lo que se espera que haga, en un tiempo esperado para nuestros usuarios.* |

### Disponibilidad

|  |  |
| --- | --- |
| *Disponibilidad* | |
| *Definición* | *Capacidad del sistema de hacer Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.* |

### Seguridad

|  |  |
| --- | --- |
| *Seguridad* | |
| *Definición* | *Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.* |

### No Repudio

|  |  |
| --- | --- |
| *No Repudio* | |
| *Definición* | *Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente* |

### Mantenibilidad

|  |  |
| --- | --- |
| *Mantenibilidad* | |
| *Definición* | *Cuál es el costo de hacer un cambio en el sistema, qué tan rápido lo puedo hacer y que tantos recursos necesito.* |

## Requerimientos Funcionales Significativos

*Los requerimientos funcionales significativos que afectan las decisiones de arquitectura son:*

|  |  |  |
| --- | --- | --- |
| *Cod.*  *Req.* | *Nombre* | *Descripción* |
| *RFS01* | *Sistema de Recomendación* | *La aplicación debe ofrecer sugerencias de outfits basadas en las últimas tendencias de moda y en el estilo personal del usuario.* |
| *RFS02* | *Planificador de Conjuntos* | *La aplicación debe permitir a los usuarios planificar conjuntos de ropa para diferentes ocasiones, como trabajo, eventos sociales, citas, etc. De acuerdo con su guardarropa.* |
| *RFS03* | *Integración con el Guardarropa* | *La aplicación debe permitir a los usuarios integrar su guardarropa actual, ya sea mediante la carga de fotos de prendas o mediante la selección de artículos de una base de datos preexistente.* |
| *RFS04* | *Personalización de Outfits por Ocasión* | *Debe permitir a los usuarios buscar outfits según la ocasión y adaptar las sugerencias según sus preferencias y requisitos específicos.* |
| *RFS05* | *Herramienta de Combinación Automática* | *Se debe implementar una herramienta de inteligencia artificial (IA) que sugiera combinaciones de prendas automáticamente, teniendo en cuenta factores como el color, el estilo y la ocasión.* |
| *RFS06* | *Notificación de inspiración y Planificación* | *La aplicación debe enviar notificaciones periódicas con sugerencias de outfits, consejos de moda y recordatorios para planificar conjuntos para eventos próximos.* |

## Requerimientos No Funcionales

* *Rendimiento:*

*La aplicación debe cargar rápidamente, con tiempos de respuesta de menos de 2 segundos para todas las acciones principales.*

*La aplicación debe ser capaz de manejar picos de tráfico sin degradación significativa del rendimiento.*

* *Disponibilidad:*

*La aplicación debe estar disponible las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad planificado no superior al 0.1% del tiempo total.*

*Se debe implementar un sistema de respaldo y recuperación para garantizar la disponibilidad continua en caso de falla del servidor principal.*

* *Seguridad:*

*Los datos de los usuarios deben estar protegidos mediante cifrado tanto en reposo como en tránsito.*

*Se debe implementar un sistema de autenticación robusto, como el uso de contraseñas seguras y la verificación de dos factores, para proteger las cuentas de usuario.*

*Se debe realizar pruebas de seguridad regulares, incluyendo pruebas de penetración, para identificar y mitigar posibles vulnerabilidades.*

* *Escalabilidad:*

*La arquitectura de la aplicación debe ser escalable horizontalmente, permitiendo añadir más recursos según sea necesario para manejar un aumento en la carga de trabajo.*

*La aplicación debe ser capaz de manejar al menos 100.000 usuarios concurrentes sin degradación significativa del rendimiento.*

* *Usabilidad:*

*La interfaz de usuario debe ser intuitiva y fácil de usar, con un diseño limpio y coherente en todas las pantallas.*

*La aplicación debe ser accesible para personas con discapacidades, cumpliendo con las pautas de accesibilidad web (WCAG).*

* *Compatibilidad:*

*La aplicación debe ser compatible con una amplia gama de dispositivos y navegadores web, incluyendo dispositivos móviles, tabletas y ordenadores de escritorio.*

*Se deben realizar pruebas de compatibilidad cruzada para garantizar un rendimiento consistente en diferentes plataformas y configuraciones de dispositivos.*

* *Mantenibilidad:*

*El código de la aplicación debe seguir prácticas de desarrollo limpias y estar bien documentado para facilitar su mantenimiento y futuras actualizaciones.*

*Se debe implementar un sistema de monitoreo y registro para facilitar la detección y resolución de problemas de manera proactiva.*

## Supuestos y Restricciones

* *Supuestos:*

*Acceso a datos de proveedores externos:*

*Se supone que la aplicación tendrá acceso a APIs de proveedores externos de moda para obtener información actualizada sobre productos, precios y disponibilidad de inventario.*

*Conexión a Internet:*

*Se supone que los usuarios tendrán acceso a una conexión a Internet estable para utilizar todas las funcionalidades de la aplicación.*

*Disponibilidad de recursos de desarrollo:*

*Se supone que el equipo de desarrollo tendrá acceso a los recursos necesarios, incluyendo hardware, software y herramientas de desarrollo, para llevar a cabo el proyecto de manera eficiente.*

*Participación de los usuarios:*

*Se supone que los usuarios participarán activamente en la aplicación, proporcionando retroalimentación y utilizando las funcionalidades ofrecidas de manera regular.*

* *Restricciones:*

*Compatibilidad con navegadores y dispositivos:*

*La aplicación estará restringida a ser compatible con una lista específica de navegadores web y dispositivos móviles para garantizar una experiencia de usuario consistente.*

*Cumplimiento normativo:*

*La aplicación estará sujeta a ciertas regulaciones y normativas, como las leyes de protección de datos y privacidad, que deben cumplirse en todo momento.*

*Limitaciones de presupuesto y tiempo:*

*El desarrollo de la aplicación estará restringido por un presupuesto y un cronograma definidos, lo que podría afectar la disponibilidad de recursos y el alcance del proyecto.*

*Disponibilidad de APIs externas:*

*La funcionalidad de la aplicación podría estar limitada al momento de implementar o hacer uso de APIs exteras, debido a posibles* *o interrupciones en los servicios proporcionados por proveedores externos de moda y otros servicios de terceros. Estos fallos pueden restringir la estabilidad y accesibilidad de la aplicación al depender de recursos externos.*

# Contexto Del Sistema

*La idea de plantear una aplicación para crear outfits personalizados viene desde la perspectiva de abordar varios problemas y necesidades que la sociedad tiene referente al mundo del uso de la moda y la vestimenta. Conceptualmente se le conoce como moda al “gusto colectivo y cambiante en lo relativo a prendas de vestir y complementos” o al “uso, modo o costumbre que está en tendencia durante algún tiempo, o en determinado país”.*

*La moda ha estado presente a lo largo del tiempo, no es algo que ha surgido recientemente, sino que viene desde la historia de la humanidad, tanto así que esta ha generado cierto impacto en la sociedad algunos de ellos son:*

* *La manera de vestir es considerada como la más grande expresión de libertad.*
* *El mero hecho de “vestir”, de cubrir nuestro cuerpo con prendas responde a un requerimiento social.*
* *Manifestación visual y tangible de la cultura, lo cual nos permite expresar quiénes somos, de dónde venimos y qué valoramos.*
* *La moda como la cultura tienen mucho que ver con el consumo, ese impulso irresistible de comprar y una vez realizada la compra producir cierto alivio de tensión. Todo lo anterior impulsando a una compra innecesaria y poco consciente.*

*Teniendo en cuenta lo explicado anteriormente podemos entender un poco que el tema de la moda es algo que abarcar mundialmente y a cada una de las personas, aunque no le encontremos un significado como tal hacemos uso diario de ella, por ello la idea de la aplicación, la cual con esta se busca encontrar soluciones a problemas comunes que motivan al desarrollo de esta aplicación. Algunos* ***problemas comunes*** *son:*

* *Falta de inspiración al momento de elegir una vestimenta, el uso de la aplicación podría ofrecer sugerencias y combinaciones basadas quizá en últimas tendencias y estilos personales, según información proporcionada por el usuario.*
* *Compras compulsivas, la aplicación ayudaría a planificar conjuntos y a su vez evitar ese tipo de compras innecesarias, fomentando compras más conscientes e integrándola con tu guardarropa existente.*
* *Tiempo y esfuerzo para combinar prendas, quizá muchos experimenten esta problemática, pero con la ayuda de la aplicación se podría simplificar este proceso, ahorrando tiempo y esfuerzo a los usuarios, quizá implementando una herramienta de IA para satisfacer esas necesidades.*
* *Problemas de almacenamiento y organización, la mayoría de las personas tienden a tener mucho cúmulo de ropa en su armario, tan así que se les hace difícil poder organizar y a su vez recordar esas opciones disponibles para crear su outfit. Con la aplicación se permitiría digitalizar el armario y así organizar de forma más simple los outfits.*
* *Diversidad de ocasiones, la aplicación podría ofrecer opciones o notificaciones de sugerencia específicas para cada situación, dando así posibles atuendos adecuados para esas situaciones teniendo presente la estación del año y el clima (trabajo, eventos sociales, etc.).*

*Se sabe que el contexto del sistema abarca varios aspectos relacionados con la moda, el estilo personal y las necesidades de los usuarios. A continuación, se detallan algunos elementos del contexto y la problemática a resolver:*

* *Contexto del Sistema*:

***Tendencias de Moda:*** *La moda es un fenómeno cultural en constante cambio, influenciado por factores como eventos sociales, celebridades y cambios en la sociedad. La aplicación debe estar al tanto de las tendencias actuales para ofrecer recomendaciones relevantes a los usuarios.*

***Diversidad de Estilos:*** *Los usuarios tienen una amplia variedad de estilos personales y preferencias de moda. La aplicación debe ser capaz de adaptarse a estos diferentes estilos y ofrecer recomendaciones personalizadas para cada usuario.*

***Acceso a Productos:*** *La aplicación depende del acceso a una amplia variedad de productos de moda, incluyendo prendas de vestir y accesorios de diferentes marcas y estilos. Se requiere integración con proveedores externos para garantizar un catálogo actualizado y diverso.*

***Interacción Social:*** *La moda es a menudo un tema de interacción social, donde los usuarios comparten sus looks y opiniones con amigos y seguidores en redes sociales. La aplicación debe facilitar esta interacción social y permitir a los usuarios compartir sus outfits fácilmente.*

* *Problemática para resolver*:

***Personalización de la Experiencia:*** *Los usuarios buscan una experiencia de moda personalizada que se adapte a sus gustos y preferencias individuales. La aplicación debe ser capaz de ofrecer recomendaciones de outfits que reflejen el estilo personal de cada usuario.*

***Acceso a Tendencias Actuales:*** *Mantenerse al día con las tendencias de moda puede ser un desafío para muchos usuarios. La aplicación debe proporcionar acceso fácil a las últimas tendencias y sugerir outfits que estén en línea con las tendencias actuales.*

***Selección de Productos Adecuados:*** *Encontrar prendas de vestir y accesorios que se ajusten al estilo personal y a las necesidades específicas de cada usuario puede ser abrumador. La aplicación debe facilitar la búsqueda y selección de productos adecuados, ofreciendo recomendaciones precisas y relevantes.*

***Interacción Social y Retroalimentación:*** *Los usuarios valoran la opinión de sus amigos y seguidores sobre sus estilos. La aplicación debe facilitar la interacción social y proporcionar herramientas para que los usuarios compartan sus outfits y reciban retroalimentación de manera rápida y fácil.*

***Acceso a una Amplia Variedad de Productos:*** *La aplicación debe ofrecer acceso a una amplia variedad de productos de moda de diferentes marcas y estilos para satisfacer las necesidades y preferencias de todos los usuarios.*

## Stack Tecnológico

* *Frontend:*

*Framework JavaScript: React.js o Angular.js*

*Lenguajes de Marcado y Estilos: HTML5 y CSS3*

*Bibliotecas de Diseño: Bootstrap*

*Gestión De Estados: Redux*

* *Backend:*

*Lenguaje de Programación: JAVA*

*Base De Datos: PostgreSQL*

*Autenticación y Autorización: JSON Web Tokens*

* *Servicios De Integración:*

*API RESTful: SpringBoot*

*Integración de Proveedores Externos: APIS de proveedores externos de moda*

* *Motor de Recomendaciones:*

*Algoritmos de Recomendación: Se podría hacer una implementación con Python utilizando bibliotecas como TensorFlow*

* *Seguridad:*

*Cifrado de Datos: Utilizando HTTPS*

*Protección contra Amenazas: CORS*

* *Infraestructura:*

*Plataforma de Nube: AWS*

*Contenedores: Docker*

# Diagramas

## Diagrama de Clases

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

***Explicación del Diagrama***

*El diagrama de clases anterior representa la estructura estática del sistema, mostrando las entidades clave y las relaciones entre ellas. En el contexto del proceso de creación de outfits personalizados, el diagrama de clases incluye las entidades relevantes, sus atributos y sus métodos, así como las relaciones entre estas entidades.*

*Aquí hago una explicación más detallada del diagrama:*

***Usuario:*** *Representa los usuarios de la aplicación que crean y utilizan outfits personalizados.*

***Prenda:*** *Representa las prendas de vestir que los usuarios agreguen en la aplicación.*

***Outfit:*** *Representa los conjuntos de prendas seleccionadas para crear outfits personalizados.*

***Evento:*** *Representa eventos específicos para los cuales los usuarios pueden crear outfits.*

***EstiloOutfit:*** *Representa los estilos de outfits que los usuarios pueden aplicar a sus conjuntos.*

***Categoría de Prenda:*** *Representa las categorías o tipos de prendas disponibles en la aplicación.*

***Clima:*** *Representa los diferentes tipos de clima que pueden influir en las sugerencias de outfits.*

***Notificación:*** *Representa las notificaciones que la aplicación puede enviar a los usuarios.*

***Sugerencia de Outfit:*** *Representa las sugerencias de outfits generadas por la aplicación para los usuarios.*

***Estación del Año:*** *Representa las estaciones del año que pueden afectar a las tendencias de moda y a los tipos de outfits recomendados.*

## Diagrama de Flujo de Datos

Diagrama

Descripción generada automáticamente

***Explicación del Diagrama***

*El diagrama de flujo de datos anterior describe el proceso sistemático para crear un outfit personalizado. En términos generales, el diagrama ilustra cómo un sistema puede tomar la información proporcionada por un usuario y, a través de una serie de pasos lógicos y filtrados, producir una recomendación de vestimenta que se ajuste a las preferencias y necesidades específicas del usuario.*

*Aquí hago la explicación más detallada del propósito y funcionamiento del diagrama:*

***Usuario:*** *Todo comienza con el usuario, quien proporciona sus datos personales y selecciona sus preferencias de estilo lo que se utilizará como base para el proceso.*

***Seleccionar Preferencias:*** *El usuario elige sus preferencias, que pueden incluir tipo de prenda, color, ocasión, etc.*

***Generar Outfit Personalizado:*** *Utilizando los datos del usuario y las preferencias seleccionadas, el sistema genera opciones de outfits personalizados.*

***Conjunto Prendas:*** *Es el conjunto de prendas seleccionadas que forman el outfit, que coinciden con los criterios del usuario.*

***Selección por Ocasión:*** *Se guarda y se aplica un filtro adicional para asegurar que el outfit recomendado sea apropiado para la ocasión específica que el usuario tiene en mente.*

***Búsqueda Del Outfit:*** *El sistema aplica un filtro basado en la ocasión o evento para el cual el usuario desea el outfit.*

***Mostar Outfit:*** *Después de aplicar el filtro, se presenta al usuario el outfit final para su aprobación o ajustes.*

*Este proceso asegura que el usuario reciba un conjunto de prendas bien coordinado y adecuado a sus necesidades y preferencias.*

## Diagrama de Paquetes

Diagrama

Descripción generada automáticamente con confianza media

***Explicación del Diagrama***

*El diagrama de paquetes es una herramienta de modelado para visualizar y organizar los elementos de un sistema de software en grupos lógicos o funcionales. Sirve para proporcionar una vista de alto nivel de la estructura de un sistema y cómo sus componentes están agrupados y relacionados entre sí, en otras palabras, nos proporciona una vista de alto nivel de la estructura del proceso para la creación de Outfits Personalizados.*

*Aquí hago la explicación más detallada del funcionamiento del diagrama:*

***Aplicación de Creación de Outfits:*** *Este paquete representa la aplicación en su conjunto. Contiene las clases y componentes que forman parte de la aplicación de creación de outfits personalizados.*

***Controladores:*** *Este paquete contiene los controladores, que son responsables de manejar las solicitudes de los usuarios y de coordinar las acciones dentro de la aplicación. Cada controlador se encarga de una parte específica de la lógica de negocio, como la gestión de usuarios, outfits, prendas, eventos, estilos, tendencias de moda, sugerencias de outfits y notificaciones.*

***Vistas:*** *Este paquete contiene las vistas, que son responsables de mostrar la interfaz de usuario al usuario final. Cada vista corresponde a una parte específica de la aplicación, como la gestión de usuarios, la visualización de outfits, la administración de prendas, la planificación de eventos, etc.*

***Servicios:***

*Este paquete contiene los servicios adicionales que la aplicación puede utilizar, como la integración con API externas (por ejemplo, para obtener datos meteorológicos) y la gestión de notificaciones. Estos servicios proporcionan funcionalidades adicionales que complementan la lógica principal de la aplicación.*

***Persistencia:***

*Este paquete contiene los objetos de acceso a datos (DAO), que son responsables de interactuar con la capa de persistencia de datos. Los DAOs proporcionan métodos para almacenar, recuperar, actualizar y eliminar datos de la base de datos. Cada DAO se encarga de una entidad específica del sistema, como usuarios, outfits, prendas, eventos, etc.*

## Diagrama de Despliegue

Texto

Descripción generada automáticamente

***Explicación del Diagrama***

*El diagrama de despliegue proporciona una alta disponibilidad, escalabilidad y separación de preocupaciones al separar cada apartado en su propio entorno de ejecución, es decir, cada uno se encarga de ejecutar el apartado que le corresponde. En este caso este diagrama no es para un proceso único como se ha venido manejando anteriormente, sino que aquí se muestra el despliegue del sistema completo.*