Trabajo Práctico NRO 2:

Planificación

Objetivos

Adquirir las competencias básicas de la planificación del trabajo en un entorno ágil.

Conocer qué metodologías existen, sus diferencias y poder reconocer cuál utilizar en un caso dado.

Practicar la comunicación por medio de la redacción de tareas (tickets) en un proyecto de software.

Temas

* Visión de Productos y Product Roadmap.
* Elección de la metodología de trabajo ágil.
* Obtención y creación de requerimientos.
* Creación de tareas y subtareas. User Stories, features, functions.
* Estimaciones.

Ejercicios

1. Una empresa llamada Sonarg quiere competir con Spotify por un lugar en el mercado de la música. Siendo el Project Manager del nuevo proyecto, te encargan crear una copia del sistema de administración de librerías de playlists de Spotify. Teniendo en cuenta que el tiempo asignado es de 3 meses y el equipo está compuesto de un diseñador y dos programadores, uno de backend y otro de frontend:
2. Crear una lista de al menos 10 requerimientos.

1. Registro de usuarios: Permitir a los usuarios registrarse con una cuenta para acceder al sistema y crear sus propias playlists.

2. Inicio de sesión seguro: Implementar un sistema de autenticación robusto para proteger la privacidad de los usuarios al iniciar sesión en sus cuentas.

3. Creación de playlists: Permitir a los usuarios crear nuevas playlists, asignarles un nombre y agregar canciones de la librería de Sonarg.

4. Edición de playlists: Brindar a los usuarios la capacidad de editar el contenido de sus playlists, incluyendo agregar, eliminar y reordenar canciones.

5. Búsqueda de canciones: Integrar una función de búsqueda para que los usuarios puedan encontrar fácilmente canciones en la librería de Sonarg y agregarlas a sus playlists.

6. Reproducción de música: Permitir a los usuarios reproducir las canciones de sus playlists directamente desde la aplicación.

7. Compartir playlists: Ofrecer la opción de compartir playlists a través de enlaces o redes sociales para que los usuarios puedan compartir sus gustos musicales con otros.

8. Seguir playlists: Permitir a los usuarios seguir las playlists creadas por otros usuarios y recibir actualizaciones cuando se agreguen nuevas canciones.

9. Compatibilidad multiplataforma: Asegurarse de que la aplicación sea compatible con una variedad de dispositivos y sistemas operativos, como dispositivos móviles, tabletas y computadoras de escritorio.

10. Diseño intuitivo y atractivo: Crear una interfaz de usuario atractiva y fácil de usar que brinde una experiencia fluida y agradable para los usuarios al navegar por la aplicación y gestionar sus playlists.

1. Crear un Roadmap con los entregables.

Mes 1: Investigación y Diseño

1. Investigación de mercado: Analizar el mercado de la música y la competencia, identificar oportunidades y amenazas.

2. Requisitos del sistema: Refinar y documentar los requisitos del sistema en colaboración con el equipo y los interesados.

3. Diseño de la interfaz de usuario: Crear prototipos y diseñar la interfaz de usuario para la aplicación web y móvil.

4. Diseño de la arquitectura del sistema: Definir la arquitectura de software y la estructura de la base de datos para el sistema.

Mes 2: Desarrollo

5. Implementación del backend: Desarrollar la lógica de negocio y la funcionalidad del backend para la gestión de usuarios, playlists y reproducción de música.

6. Implementación del frontend: Desarrollar la interfaz de usuario interactiva y responsiva para la aplicación web y móvil.

7. Integración de API de música: Integrar APIs de servicios de música para acceder al catálogo de canciones y permitir la reproducción de música en la aplicación.

8. Pruebas de unidad y de integración: Realizar pruebas exhaustivas para garantizar el funcionamiento correcto y la integración sin problemas de los componentes del sistema.

Mes 3: Lanzamiento y Mejoras

9. Pruebas de usuario: Realizar pruebas de usuario para obtener retroalimentación sobre la usabilidad y la experiencia del usuario.

10. Corrección de errores y optimización: Resolver cualquier problema identificado durante las pruebas y optimizar el rendimiento del sistema.

11. Lanzamiento suave: Implementar el sistema en un entorno de producción limitado para una audiencia seleccionada.

12. Lanzamiento completo: Lanzar el sistema de forma completa para el público en general, promocionando la aplicación y sus características.

13. Monitorización y retroalimentación: Supervisar el rendimiento del sistema en producción y recopilar comentarios de los usuarios para futuras mejoras.

Este roadmap proporciona una guía general para el desarrollo del proyecto, dividiendo las tareas en etapas claramente definidas y asignando plazos realistas para cada una.

1. Elegir el stack de tecnología a utilizar y justificarlo.

1. Lenguaje de programación:

- Backend: Python

- Frontend: JavaScript (ReactJS para la web y React Native para móviles)

2. Framework de desarrollo:

- Backend: Django (para la rapidez en el desarrollo, la seguridad y la robustez)

- Frontend: React (para una interfaz de usuario dinámica y fácil de mantener)

3. Base de datos:

- PostgreSQL (por su robustez, escalabilidad y capacidades avanzadas de gestión de datos)

4. Servicios de música:

- API de Spotify (para acceder al catálogo de música y permitir la reproducción de canciones)

Justificación:

- Python y Django: Python es un lenguaje de programación versátil, con una gran cantidad de bibliotecas y frameworks disponibles. Django es un framework web de alto nivel que proporciona una gran cantidad de herramientas y funcionalidades listas para usar, lo que acelera el desarrollo y garantiza la seguridad del sistema.

- React y React Native: React es una biblioteca JavaScript para construir interfaces de usuario interactivas. Utilizar React para el frontend permite crear una experiencia de usuario moderna y receptiva tanto en la web como en dispositivos móviles, con la capacidad de compartir código entre las dos plataformas gracias a React Native.

- PostgreSQL: PostgreSQL es una base de datos relacional de código abierto muy potente y confiable. Ofrece características avanzadas como soporte para tipos de datos complejos, integridad referencial, replicación y escalabilidad, lo que lo convierte en una opción sólida para el almacenamiento de datos en aplicaciones empresariales.

- API de Spotify: Utilizar la API de Spotify permite acceder a un amplio catálogo de música y funcionalidades relacionadas, lo que facilita la integración de contenido musical en la aplicación de Sonarg y proporciona una experiencia de usuario completa y enriquecedora.

En resumen, este stack de tecnología proporciona una combinación sólida de herramientas y tecnologías probadas, que permitirán desarrollar un sistema de administración de librerías de playlists de Sonarg de manera eficiente, segura y con una experiencia de usuario de alta calidad.

1. Crear un backlog con los tickets de cada requerimiento. Cada ticket debe incluir título, descripción, requerimientos funcionales y no funcionales, pasos para realizar las pruebas y diseños en los casos que sea necesario.

1. Registro de usuarios:

Ticket 1: Crear página de registro

- Descripción: Diseñar e implementar la página de registro donde los usuarios puedan crear una cuenta en Sonarg.

- Requerimientos funcionales:

- Los usuarios deben poder ingresar su nombre de usuario, correo electrónico y contraseña.

- Validar la fortaleza de la contraseña ingresada.

- Verificar que el correo electrónico ingresado sea único en el sistema.

- Requerimientos no funcionales:

- La página debe ser intuitiva y fácil de usar.

- Debe ser compatible con dispositivos móviles y de escritorio.

- Pasos para realizar las pruebas:

1. Acceder a la página de registro.

2. Ingresar datos válidos en los campos requeridos.

3. Verificar que se cree la cuenta correctamente.

- Diseño: Se adjunta diseño de la página de registro.

2. Inicio de sesión seguro:

Ticket 2: Implementar autenticación de usuarios

- Descripción: Desarrollar el sistema de autenticación para que los usuarios puedan iniciar sesión en sus cuentas.

- Requerimientos funcionales:

- Los usuarios deben poder iniciar sesión con su nombre de usuario o correo electrónico y su contraseña.

- Validar que los datos ingresados sean correctos y coincidan con los registros en la base de datos.

- Requerimientos no funcionales:

- La autenticación debe ser segura, utilizando técnicas como hash de contraseñas.

- La página de inicio de sesión debe ser accesible desde cualquier dispositivo.

- Pasos para realizar las pruebas:

1. Acceder a la página de inicio de sesión.

2. Ingresar credenciales válidas.

3. Verificar que se inicie sesión correctamente.

- Diseño: Se adjunta diseño de la página de inicio de sesión.

3. Creación de playlists:

Ticket 3: Implementar función de creación de playlists

- Descripción: Desarrollar la funcionalidad que permita a los usuarios crear nuevas playlists en Sonarg.

- Requerimientos funcionales:

- Los usuarios deben poder asignar un nombre a la playlist.

- Deben poder agregar canciones a la playlist desde la librería de Sonarg.

- Requerimientos no funcionales:

- La funcionalidad debe ser fácil de encontrar y usar.

- Debe tener capacidad para gestionar un gran número de playlists por usuario.

- Pasos para realizar las pruebas:

1. Crear una nueva playlist.

2. Asignar un nombre a la playlist.

3. Agregar canciones a la playlist.

- Diseño: Se adjunta diseño de la página de creación de playlists.

4. Agregar canciones:

Ticket 4: Agregar canciones a playlists

-Descripción: Desarrollar la funcionalidad que permita a los usuarios buscar canciones en el catálogo de Sonarg y agregarlas a sus playlists.

-Requerimientos funcionales:

-Los usuarios deben poder buscar canciones por título, artista, álbum, etc.

-Seleccionar una canción de los resultados de búsqueda.

-Agregar la canción seleccionada a una playlist existente.

-Requerimientos no funcionales:

-La búsqueda de canciones debe ser rápida y eficiente.

-Pasos para realizar las pruebas:

-Ir a la sección de búsqueda de canciones.

-Realizar una búsqueda utilizando diferentes criterios.

-Seleccionar una canción de los resultados de búsqueda.

-Verificar que se pueda agregar la canción a una playlist existente del usuario.

-Diseños: Se puede incluir un diseño de la interfaz de búsqueda de canciones y de la interfaz para agregar canciones a una playlist.

5. Edición de playlists:

Ticket 5: Agregar función de edición de playlists

- Descripción: Implementar la capacidad de editar el contenido de las playlists existentes.

- Requerimientos funcionales:

- Los usuarios deben poder agregar, eliminar y reordenar canciones en una playlist.

- Deben poder cambiar el nombre de la playlist.

- Requerimientos no funcionales:

- La función de edición debe ser intuitiva y fácil de usar.

- Debe funcionar de manera eficiente incluso con playlists grandes.

- Pasos para realizar las pruebas:

1. Abrir una playlist existente.

2. Realizar cambios en el contenido de la playlist (agregar, eliminar, reordenar canciones).

3. Guardar los cambios y verificar que se reflejen correctamente.

- Diseño: Se adjunta diseño de la página de edición de playlists.

6. Reproducción de música

Ticket 6: Implementar reproducción de música

- Descripción: Desarrollar la funcionalidad que permita a los usuarios reproducir las canciones de sus playlists directamente desde Sonarg.

-Requerimientos funcionales:

-Los usuarios deben poder iniciar y detener la reproducción de una canción.

-Controlar el volumen y la reproducción (pausa, avance rápido, retroceso).

-Requerimientos no funcionales:

-La reproducción de música debe ser fluida y sin interrupciones.

-Pasos para realizar las pruebas:

-Abrir una playlist y seleccionar una canción.

-Iniciar la reproducción de la canción.

-Probar las funcionalidades de control de reproducción.

-Verificar que la reproducción sea fluida y sin errores.

-Diseños: Se puede incluir un diseño del reproductor de música.

7. Compartir playlists

Ticket 7: Implementar sistema de compartir playlists

-Descripción: Desarrollar la funcionalidad que permita a los usuarios compartir sus playlists con otros usuarios de Sonarg a través de enlaces directos o redes sociales.

-Requerimientos funcionales:

-Generar un enlace único para cada playlist que se pueda compartir.

-Integrar botones de compartir en redes sociales (por ejemplo, Facebook, Twitter).

-Requerimientos no funcionales:

-La generación de enlaces y la integración con redes sociales deben ser rápidas y eficientes.

-Pasos para realizar las pruebas:

-Ir a una playlist que se desea compartir.

-Generar el enlace de compartición.

-Compartir el enlace en una red social.

-Verificar que otros usuarios puedan acceder a la playlist a través del enlace compartido.

-Diseños: Puede incluirse un diseño de los botones de compartir y del diálogo de generación de enlaces.

8.Implementar recomendaciones

Ticket 8: Implementar sistema de recomendaciones

-Descripción: Desarrollar un sistema de recomendaciones que sugiera canciones y playlists basadas en los gustos musicales y el historial de reproducción del usuario.

-Requerimientos funcionales:

-Analizar el historial de reproducción del usuario.

-Generar recomendaciones personalizadas de canciones y playlists.

-Mostrar las recomendaciones en la interfaz de usuario.

-Requerimientos no funcionales:

-El sistema de recomendaciones debe ser preciso y relevante para mejorar la experiencia del usuario.

-Pasos para realizar las pruebas:

-Reproducir varias canciones y añadirlas a playlists.

-Verificar que las recomendaciones se actualicen basadas en el historial de reproducción.

-Explorar las recomendaciones y evaluar su relevancia.

-Diseños: Se puede incluir un diseño de la sección de recomendaciones en la interfaz de usuario.

1. Crear un conjunto de pruebas de aceptación.

**Pruebas de aceptación:**

**1. Registro e inicio de sesión:**

* **Escenario 1: Registro de nuevo usuario**
  + Paso 1: Abrir la página de registro.
  + Paso 2: Completar el formulario de registro con información válida.
  + Paso 3: Verificar que se muestre un mensaje de confirmación y redirija al usuario a la página de inicio de sesión.
  + Paso 4: Iniciar sesión con las credenciales recién registradas.
  + Resultado esperado: El usuario puede iniciar sesión correctamente después del registro.

**2. Gestión de playlists:**

* **Escenario 2: Creación de nueva playlist**
  + Paso 1: Ir a la sección de playlists.
  + Paso 2: Hacer clic en el botón de "Crear nueva playlist".
  + Paso 3: Completar el formulario con un nombre único para la playlist.
  + Paso 4: Verificar que la nueva playlist se agregue correctamente a la lista de playlists del usuario.
  + Resultado esperado: Se crea una nueva playlist con éxito y se muestra en la lista de playlists del usuario.
* **Escenario 3: Agregar y eliminar canciones de una playlist**
  + Paso 1: Abrir una playlist existente.
  + Paso 2: Agregar una canción desde la biblioteca de música a la playlist.
  + Paso 3: Verificar que la canción se agregue correctamente a la playlist.
  + Paso 4: Eliminar una canción de la playlist.
  + Paso 5: Verificar que la canción se elimine correctamente de la playlist.
  + Resultado esperado: Se pueden agregar y eliminar canciones de una playlist sin errores.

**3. Reproducción de música:**

* **Escenario 4: Reproducción de una canción desde una playlist**
  + Paso 1: Abrir una playlist con al menos una canción.
  + Paso 2: Seleccionar una canción para reproducir.
  + Paso 3: Iniciar la reproducción de la canción.
  + Paso 4: Probar las funcionalidades de control de reproducción (pausa, avance rápido, retroceso).
  + Resultado esperado: La canción se reproduce correctamente y se pueden controlar las funciones de reproducción sin errores.

**4. Compartir playlists:**

* **Escenario 5: Compartir una playlist a través de enlace**
  + Paso 1: Abrir una playlist que se desea compartir.
  + Paso 2: Generar el enlace de compartición.
  + Paso 3: Compartir el enlace con otro usuario.
  + Paso 4: Verificar que el otro usuario pueda acceder a la playlist a través del enlace compartido.
  + Resultado esperado: El enlace de compartición funciona correctamente y otros usuarios pueden acceder a la playlist compartida.

**5. Sistema de recomendaciones:**

* **Escenario 6: Visualización de recomendaciones personalizadas**
  + Paso 1: Reproducir varias canciones y añadirlas a playlists.
  + Paso 2: Verificar que las recomendaciones se actualicen basadas en el historial de reproducción.
  + Paso 3: Explorar las recomendaciones y evaluar su relevancia.
  + Resultado esperado: Las recomendaciones se generan correctamente basadas en el historial de reproducción del usuario y son relevantes para sus gustos musicales.

Estas pruebas de aceptación cubren los principales flujos de usuario en Sonarg, desde el registro e inicio de sesión hasta la gestión de playlists, reproducción de música, compartición de playlists y sistema de recomendaciones. Cada escenario describe los pasos a seguir y el resultado esperado para verificar que la funcionalidad de la aplicación cumple con los requerimientos establecidos.

1. Clasificar los requerimientos en Epics.

**1. Gestión de Usuarios:**

* Registro de usuarios.
* Inicio de sesión.
* Gestión de perfiles de usuario.
* Seguridad y privacidad de la cuenta.

**2. Gestión de Playlists:**

* Creación de playlists.
* Edición de playlists.
* Eliminación de playlists.
* Compartir playlists.

**3. Exploración y Reproducción de Música:**

* Búsqueda y exploración de música.
* Reproducción de canciones.
* Control de reproducción de canciones (pausa, avance rápido, retroceso).

**4. Sistema de Recomendaciones:**

* Análisis de historial de reproducción.
* Generación de recomendaciones personalizadas.
* Visualización de recomendaciones en la interfaz de usuario.

**5. Interacción Social:**

* Compartir actividades musicales en redes sociales.
* Conexión con amigos y seguimiento de sus actividades musicales.

1. Estimar los tickets usando planning poker y agregar dicha estimación a los tickets.

**1. Registro e inicio de sesión:**

* Ticket 1: Crear sistema de registro de usuario
  + Estimación: 3 puntos
* Ticket 2: Implementar sistema de inicio de sesión
  + Estimación: 2 puntos

**2. Gestión de playlists:**

* Ticket 3: Implementar creación de nuevas playlists
  + Estimación: 5 puntos
* Ticket 4: Agregar canciones a playlists
  + Estimación: 3 puntos

**3. Edición de playlists:**

* Ticket 5: Implementar edición de playlists
  + Estimación: 5 puntos
* Ticket 6: Implementar reproducción de música
  + Estimación: 8 puntos

**4. Compartir playlists:**

* Ticket 7: Implementar sistema de compartir playlists
  + Estimación: 3 puntos
* Ticket 8: Implementar sistema de recomendaciones
  + Estimación: 8 puntos

1. Crea una tabla comparativa de las siguientes metodologías de trabajo: eXtreme Programming, Scrum, Kanban y Waterfall. Debes comparar: Filosofía, roles claves, planificación, descripción de tareas, entregables, ceremonias, flexibilidad, enfoque de calidad, entorno adecuado.
2. Crear una tabla comparativa de User Stories, Feature y Function.
3. Para el proyecto del ejercicio 1 eliger una metodología de trabajo y justificar la elección

Decidimos que la metodología de trabajo SCRUM para el proyecto de desarrollo del sistema de administración de librerías de playlists de Sonarg es una elección sólida por varias razones:

1. Iterativo e Incremental: SCRUM se basa en ciclos de desarrollo iterativos e incrementales, lo que permite obtener rápidamente versiones funcionales del producto que se pueden revisar y mejorar en cada iteración. Esto es especialmente útil en un proyecto donde se necesita entregar valor de manera temprana y continua.

2. Flexibilidad: SCRUM nos proporciona flexibilidad para adaptarnos a los cambios en los requisitos del proyecto. Al dividir el trabajo en sprints cortos y enfocarse en objetivos concretos, el equipo puede ajustar fácilmente la dirección del proyecto según las necesidades del cliente o los cambios en el mercado.

3. Colaboración y Comunicación: esta manera de trabajo fomenta la colaboración y la comunicación continua entre todos los miembros del equipo, así como con los interesados. Las reuniones diarias (daily), las reuniones de planificación de sprint y las retrospectivas permiten mantener a todos en el mismo camino y abordar los problemas de manera proactiva.

4. Entrega Continua de Valor: El equipo va a realizar entrega incrementos de valor en cada sprint, lo que permite obtener retroalimentación temprana de los interesados y ajustar el producto según sea necesario. Esto garantiza que el producto final satisfaga las necesidades reales del cliente y del mercado.

5. Transparencia y Visibilidad: Scrum promueve la transparencia y la visibilidad en el progreso del trabajo a través de artefactos como el backlog del producto, el backlog del sprint y el burndown chart. Esto ayuda a mantener a todos informados sobre el estado del proyecto y a identificar rápidamente cualquier problema o riesgo.

RESOLVER

¿Cómo mejora el machine learning los modelos de arquitectura empresarial?

El Machine Learning (ML) puede mejorar los modelos de arquitectura empresarial de varias maneras, proporcionando capacidades avanzadas de análisis, predicción y optimización. Aquí hay algunas formas en las que el ML puede mejorar los modelos de arquitectura empresarial:

1. **Análisis de datos complejos:** El ML puede procesar grandes volúmenes de datos estructurados y no estructurados para identificar patrones, tendencias y relaciones que pueden no ser evidentes para los analistas humanos. Esto puede ayudar a comprender mejor el entorno empresarial y las interacciones entre los diferentes componentes de la arquitectura.
2. **Predicción y pronóstico:** Los algoritmos de ML pueden predecir el comportamiento futuro de los sistemas empresariales, como la demanda de productos, el rendimiento financiero o el comportamiento del mercado. Estas predicciones pueden ayudar a tomar decisiones informadas y planificar estrategias futuras.
3. **Optimización de recursos:** El ML puede optimizar el uso de recursos empresariales, como la asignación de personal, la gestión de inventario o la planificación de la cadena de suministro. Al analizar datos históricos y en tiempo real, los algoritmos de ML pueden identificar patrones de uso y eficiencia, ayudando a mejorar la asignación de recursos y reducir costos.
4. **Automatización de procesos:** Mediante el uso de técnicas de ML, es posible automatizar procesos empresariales complejos, como la clasificación de correos electrónicos, la detección de fraudes o la optimización de la atención al cliente. Esto puede aumentar la eficiencia operativa y liberar recursos para tareas más estratégicas.
5. **Personalización y recomendaciones:** El ML puede mejorar la experiencia del cliente al proporcionar recomendaciones personalizadas y contenido relevante en función del comportamiento pasado y las preferencias individuales. Esto puede aumentar la satisfacción del cliente y las tasas de conversión.
6. **Detección de anomalías y seguridad:** Los algoritmos de ML pueden identificar anomalías y patrones sospechosos en los datos empresariales, lo que ayuda a detectar y prevenir actividades fraudulentas, intrusiones de seguridad o problemas operativos antes de que causen daños significativos.

En resumen, el ML puede mejorar significativamente los modelos de arquitectura empresarial al proporcionar capacidades avanzadas de análisis, predicción y optimización que ayudan a tomar decisiones más informadas, eficientes y estratégicas.

¿Cómo impulsará la inteligencia artificial a la arquitectura empresarial?

La inteligencia artificial (IA) tiene el potencial de transformar profundamente la arquitectura empresarial al proporcionar capacidades avanzadas de análisis, automatización y optimización. Aquí hay algunas formas en las que la IA impulsará la arquitectura empresarial:

1. **Automatización de procesos empresariales:** La IA puede automatizar una amplia gama de procesos empresariales, desde la gestión de inventario y la atención al cliente hasta la optimización de la cadena de suministro y la contabilidad. Esto puede aumentar la eficiencia operativa, reducir los errores humanos y liberar tiempo y recursos para tareas más estratégicas.
2. **Análisis avanzado de datos:** La IA puede analizar grandes volúmenes de datos estructurados y no estructurados para identificar patrones, tendencias y relaciones complejas que pueden no ser evidentes para los analistas humanos. Esto puede ayudar a las empresas a tomar decisiones más informadas y predecir tendencias futuras con mayor precisión.
3. **Personalización de la experiencia del cliente:** La IA puede utilizar datos de clientes para proporcionar experiencias personalizadas y relevantes en tiempo real. Esto puede incluir recomendaciones de productos, comunicaciones personalizadas y servicios adaptados a las necesidades individuales de cada cliente, lo que mejora la satisfacción del cliente y aumenta la lealtad a la marca.
4. **Optimización de recursos y procesos:** La IA puede optimizar el uso de recursos empresariales, como la asignación de personal, la gestión de inventario y la planificación de la cadena de suministro. Al analizar datos históricos y en tiempo real, la IA puede identificar patrones de uso y eficiencia, ayudando a mejorar la asignación de recursos y reducir costos.
5. **Detección de anomalías y seguridad:** La IA puede identificar anomalías y patrones sospechosos en los datos empresariales, lo que ayuda a detectar y prevenir actividades fraudulentas, intrusiones de seguridad o problemas operativos antes de que causen daños significativos.
6. **Innovación y desarrollo de productos:** La IA puede impulsar la innovación al ayudar a las empresas a desarrollar productos y servicios nuevos y mejorados. Esto puede incluir el análisis de mercado, la identificación de oportunidades de mejora y la generación de ideas creativas para productos y servicios innovadores.

En resumen, la inteligencia artificial tiene el potencial de impulsar la arquitectura empresarial al proporcionar capacidades avanzadas de análisis, automatización y optimización que pueden mejorar la eficiencia operativa, aumentar la satisfacción del cliente, reducir costos y fomentar la innovación.

¿Cómo aprender del machine learning que se ejecuta en la producción?

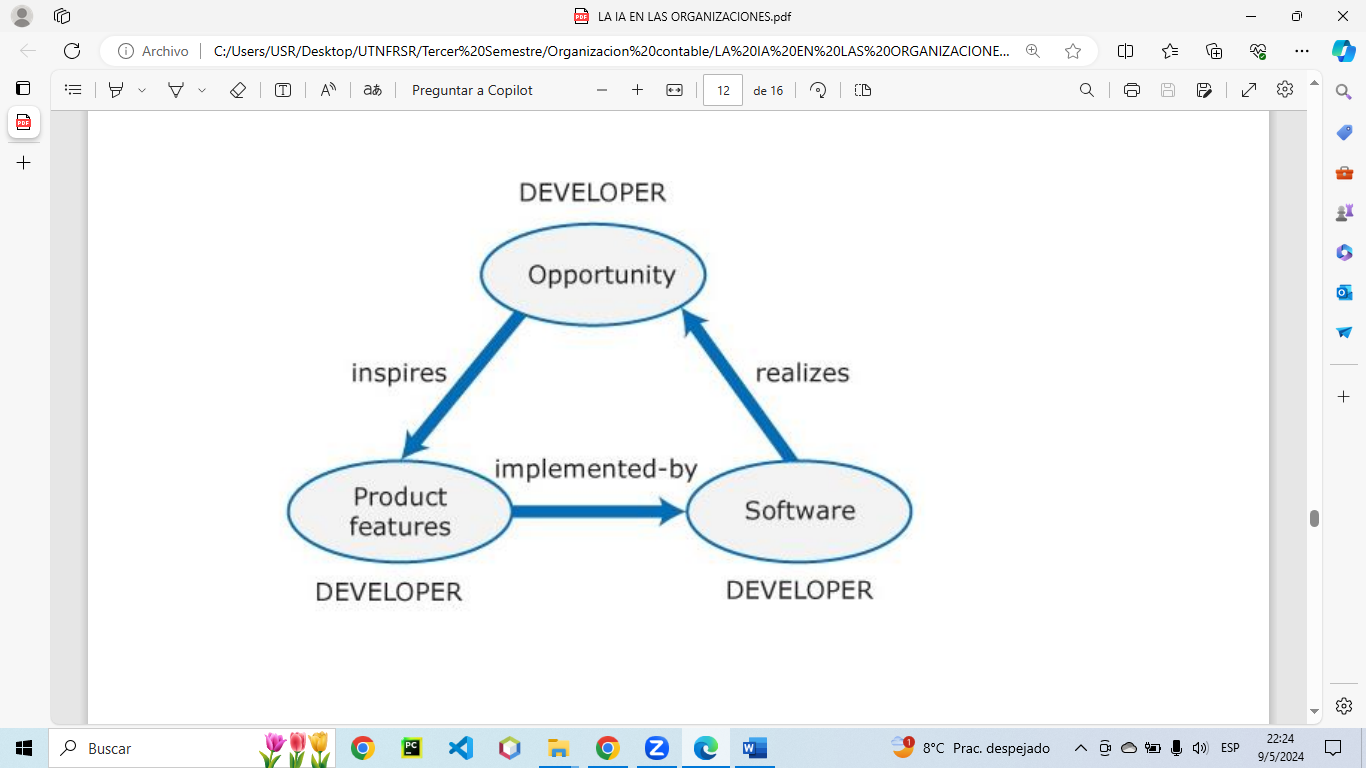
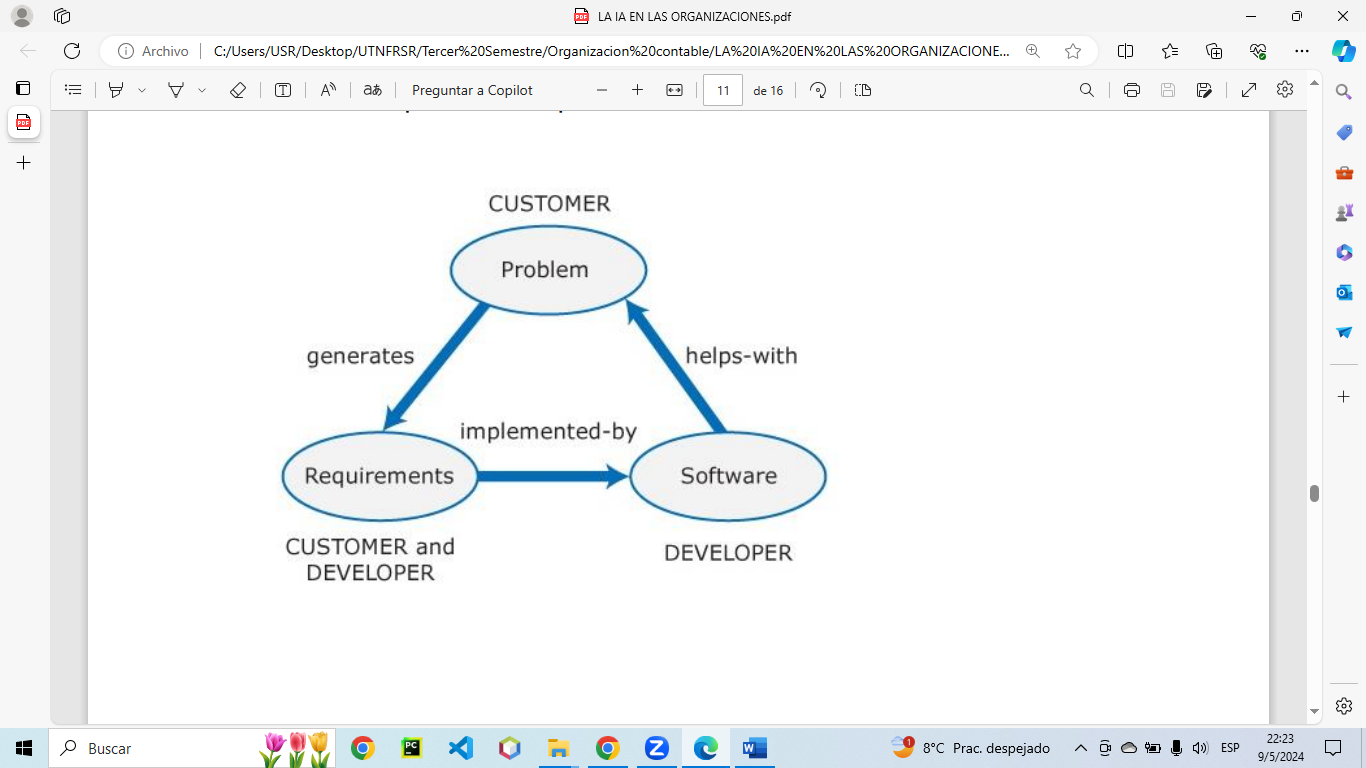
Aprender del Machine Learning (ML) que se ejecuta en producción es fundamental para mejorar continuamente los modelos y maximizar su efectividad. Aquí hay algunas prácticas clave para aprender del ML en producción:

1. **Monitoreo continuo:** Establece un sistema de monitoreo continuo para rastrear el rendimiento del modelo en producción. Esto incluye métricas como la precisión, el sesgo, la sensibilidad y la especificidad. Identifica cualquier desviación significativa en estas métricas que pueda indicar un deterioro en el rendimiento del modelo.
2. **Registro de datos:** Registra y almacena datos de entrada y salida del modelo en producción. Esto permite realizar un análisis retrospectivo para comprender cómo el modelo está tomando decisiones y cómo sus predicciones se comparan con los resultados reales.
3. **Análisis de errores:** Analiza los errores cometidos por el modelo en producción para identificar patrones y tendencias. Examina los casos en los que el modelo falló para entender las razones subyacentes y buscar oportunidades de mejora.
4. **Feedback del usuario:** Recopila comentarios y retroalimentación de los usuarios finales sobre la precisión y utilidad del modelo en producción. Utiliza esta información para identificar áreas de mejora y priorizar los cambios necesarios.
5. **Experimentación continua:** Realiza experimentos controlados para evaluar el impacto de posibles mejoras en el modelo en producción. Esto puede incluir ajustes en los hiperparámetros del modelo, la adición de nuevas características o la actualización del conjunto de datos de entrenamiento.
6. **Reentrenamiento periódico:** Establece un proceso automatizado para reentrenar periódicamente el modelo en función de nuevos datos disponibles. Esto garantiza que el modelo se mantenga actualizado y pueda adaptarse a cambios en el entorno empresarial o en el comportamiento del usuario.
7. **Despliegue gradual de cambios:** Implementa cambios en el modelo de manera gradual, utilizando técnicas como la implementación de canarios o la división de tráfico. Esto minimiza el riesgo de impacto negativo en la producción y permite evaluar el rendimiento de los cambios en condiciones reales.
8. **Colaboración interdisciplinaria:** Fomenta la colaboración entre equipos de desarrollo, científicos de datos, ingenieros de ML y usuarios finales para aprender de manera efectiva del ML en producción. Comparte conocimientos y experiencias para identificar mejores prácticas y enfoques innovadores.

Al implementar estas prácticas, puedes aprender de manera efectiva del Machine Learning que se ejecuta en producción, mejorar continuamente los modelos y garantizar que sigan siendo útiles y efectivos en un entorno empresarial dinámico.

RESOLVER: EXPLICAR EL SIGUIENTE CONCEPTO INCLUIDO EL GRAFICO

When software products were first developed, they were delivered on a disk and installed by costumers on their computers. The software ran on those computers and user data were stored on them. There was on communication between the users' computers and the vendor's computer. Now, customers can download products from either an app store or the vendor's website. Some products are still based on a stand-alone execution model in which all computation is carried out on the product owner's computers. However, ubiquitous high-speed networking means that alternative execution models are now available. In these models, the product owner's computers act as a client, whit some or all execution and data storage on the vendor's servers



En los primeros días del desarrollo de productos de software, estos se entregaban a los clientes en forma de discos físicos, quienes los instalaban en sus propias computadoras. El software funcionaba en estos sistemas locales, y los datos de los usuarios se almacenaban en sus dispositivos. En ese momento, no existía comunicación entre las computadoras de los usuarios y las de los proveedores del software. Sin embargo, con el avance tecnológico, los clientes ahora pueden descargar productos directamente desde tiendas de aplicaciones o desde los sitios web de los proveedores.

Aunque algunos productos todavía se basan en un modelo de ejecución independiente, donde toda la computación se realiza en las computadoras de los propietarios, la amplia disponibilidad de redes de alta velocidad ha dado lugar a nuevos modelos de ejecución. En estos modelos, las computadoras del propietario del producto actúan como clientes, mientras que parte o la totalidad de la ejecución y el almacenamiento de datos se realizan en los servidores del proveedor, a menudo en la nube.

Esta evolución del modelo de ejecución de software ha sido posible gracias a la proliferación de servicios en la nube y al aumento de la velocidad de las conexiones a Internet. Ahora, los usuarios pueden acceder a sus aplicaciones y datos desde cualquier lugar y dispositivo, lo que ha revolucionado la forma en que interactúan con el software y los servicios en línea.

Principio del formulario