

HITO 1

APOYO A PROCESOS EMPRESARIALES

MARCOS JODAR GÓMEZ

19 DE FEBRERO DE 2024

ÍNDICE

DESCRIPCIÓN DEL PROYECTO	3
PROCESOS	4
DATOS E INFORMACIÓN	5
DATOS Y HERRAMIENTAS	6
PERSONAS Y ORGANIZACIONES	8
IMPLICACIONES DEL USO DE DATOS	9
TIPOS DE IMPLICACIONES	12
AMENAZAS COMUNES Y MEDIDAS DE MITIGACIÓN	12
IMPACTO DEL USO DE DATOS E INFORMACIÓN	13
TECNOLOGÍAS	15
TECNOLOGÍAS PARA APOYAR LOS PROCESOS DE NEGOCIO E INFORMAR LAS DECISIONES	17
BENEFICIOS DE UTILIZAR LA CIENCIA DE DATOS PARA RESOLVER PROBLEMAS	18
USO DE TÉCNICAS	21
SOFTWARE	24
DISEÑO DE SOLUCIÓN	24
IMPLEMENTAR SOLUCIÓN	32
DESARROLLO DE LA SOLUCIÓN	33
BACK-END	34
EVALUACIÓN	52
CONCLUSIÓN	58
BIBLIOGRAFÍA	59

DESCRIPCIÓN DEL PROYECTO

Mi proyecto se centra en proporcionar un sólido “Apoyo a Procesos Empresariales”. En esencia, busca facilitar y mejorar la ejecución de los procesos empresariales. A través de esta iniciativa, estamos enfocados en ofrecer herramientas y recursos que optimicen la eficiencia operativa, permitiendo una gestión más eficaz y fluida de las actividades empresariales clave. Mi meta es simplificar las operaciones empresariales y proporcionar soluciones prácticas para potenciar el rendimiento general de las empresas.

PROCESOS

Proceso Empresarial:

Son las **actividades** fundamentales que la empresa realiza para **cumplir con sus objetivos** y ofrecer **valor a sus clientes**. Estos procesos son principales para la misión de la empresa, estando directamente relacionados con la producción y entrega de productos o servicios.

Proceso de Apoyo:

Son actividades que, aunque no son las principales, son esenciales para el funcionamiento eficiente de la empresa. Incluyen **funciones** como la gestión de *recursos humanos*, *contabilidad*, *tecnologías de la información*, entre otros. Estos procesos contribuyen al éxito de la empresa.

Diferenciación entre Proceso Empresarial y Proceso de Apoyo

Los Datos e Información facilitan estos Procesos:

- Proceso Empresarial:

- Los datos sobre la demanda del mercado pueden informar las estrategias de producción y marketing.
- La retroalimentación de los clientes proporciona información valiosa para mejorar productos y servicios.

- Proceso de Apoyo:

- Los datos de recursos humanos facilitan la gestión del personal y la toma de decisiones relacionadas con el talento.
- La información financiera es esencial para la contabilidad y la planificación financiera.

Herramientas para Obtener Datos:

- **Sistemas de Gestión Empresarial (ERP):** Integran datos y procesos de diferentes departamentos, proporcionando una perspectiva global de la empresa.
- **Analítica Empresarial:** Herramientas como tableros de control y análisis ayudan a interpretar datos y coger conocimientos para la toma de decisiones.
- **Encuestas y Retroalimentación del Cliente:** Plataformas para recopilar opiniones y comentarios que informan sobre la satisfacción del cliente y otras mejoras.

Lo aplicamos a los Procesos:

- **Proceso Empresarial:** Empresa de fabricación, la implementación de sensores en la cadena de producción recopila datos en tiempo real sobre la eficiencia y calidad del producto, mejorando en la toma de decisiones, optimizando la producción.
- **Proceso de Apoyo:** En el departamento de recursos humanos, el uso de sistemas de gestión de talento, recopilan datos sobre el desempeño de los empleados y sobre las tendencias del personal, facilitando la toma de decisiones estratégicas en la gestión del talento.

La correcta gestión de datos e información es muy importante para potenciar tanto los procesos empresariales como los procesos de apoyo, contribuyendo al éxito de la empresa.

.....

DATOS E INFORMACIÓN

.....

Los datos y la información desempeñan un papel crucial en el respaldo y optimización de los procesos empresariales, generando un interés significativo para las organizaciones actuales.

Estos **datos e información** se centran en los **procesos empresariales**:

1. Información para comunicar en la Toma de Decisiones:
Procesos Estratégicos: Los datos proporcionan información valiosa para la toma de decisiones estratégicas. *Analizar tendencias del mercado, comportamiento del cliente y desempeño interno* ayuda a las organizaciones a adaptarse y anticiparse a cambios.
2. Eficiencia Operativa a través de la Automatización:
Automatización de Procesos: Los datos facilitan la *automatización de procesos*, mejorando la eficiencia operativa. La *recopilación y análisis de datos* identifica áreas para la automatización, reduciendo errores y aumentando la velocidad de ejecución.

3. Personalización y Mejora Continua:

- **Experiencia del Cliente:** La información sobre *preferencias y comportamientos de los clientes* permite personalizar productos y servicios, mejorando su experiencia. La retroalimentación constante a través de datos, mejora la oferta de productos y servicios.

4. Gestión de Recursos y Optimización Financiera:

- **Gestión de Recursos Humanos:** Datos sobre el desempeño del personal, la satisfacción laboral y la retención, ayudan en la gestión de recursos humanos. Ayudando a la creación de equipos eficientes y comprometidos.
- **Optimización Financiera:** La *información financiera* respalda la planificación presupuestaria, el seguimiento del rendimiento financiero y la identificación de oportunidades. Esto mejora la rentabilidad.

5. Cumplimiento Normativo y Mitigación de Riesgos:

- **Gestión de Riesgos:** Los datos identifican y mitigan riesgos empresariales. La gestión de riesgos con datos, permite evaluar amenazas potenciales y tomar medidas proactivas.

Valor para las Organizaciones:

- **Competitividad:** La capacidad para *aprovechar datos* de manera efectiva otorga una *ventaja competitiva*. Las organizaciones que pueden adaptarse rápidamente a cambios, tienen una posición más sólida en el mercado.
- **Innovación:** La innovación se impulsa mediante la *exploración y el análisis de datos*. La identificación de nuevas oportunidades y la creación de soluciones innovadoras se basan en la comprensión de los datos.

Los datos y la información no solo respaldan los procesos empresariales, sino que también **representan un activo estratégico**. Impulsan la eficiencia, la personalización, la toma de decisiones informadas, además de la innovación, convierte a los datos en un recurso muy importante para las organizaciones en la búsqueda de la excelencia operativa y la ventaja competitiva.

.....

DATOS Y HERRAMIENTAS

.....

La creación de datos es un proceso que involucra distintas fuentes y actividades, y su manipulación es importante, tanto para convertirlos en información significativa que pueda apoyar operaciones empresariales.

Para generar datos y utilizar herramientas para su manipulación para formar datos significativos, te mostraré algunos ejemplos:

Generación de Datos:

La mayoría de las empresas registran datos a través de **SISTEMAS TRANSACCIONALES** (para procesar transacciones comerciales), como *sistemas de ventas, gestión de inventario y plataformas de relaciones con clientes (CRM)*. Cada transacción genera datos relevantes sobre las actividades comerciales.

En entornos modernos, **SENSORES Y DISPOSITIVOS CONECTADOS A INTERNET (IoT)** recopilan datos en tiempo real. Como los sensores en la cadena de suministro, dispositivos de seguimiento de flotas y monitoreo ambiental.

Las **PLATAFORMAS EN LÍNEA**, como *sitios web y aplicaciones móviles*, generan datos a través de **interacciones de usuarios**. Estos datos pueden incluir comportamientos de navegación, preferencias de productos y comentarios.

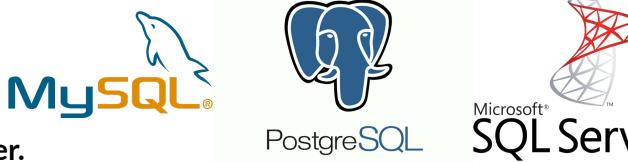
LAS REDES SOCIALES generan muchísimos datos significativos. Con los *Comentarios, likes o "me gusta"*, *compartidos* y otras interacciones proporcionan información sobre la percepción del cliente y las tendencias del mercado.

Herramientas para Manipular Datos:

- **BASES DE DATOS RELACIONALES:** Utilizadas para almacenar y organizar grandes conjuntos de datos.

Ejemplos:

- MySQL
- PostgreSQL
- Microsoft SQL Server.



- **Herramientas de EXTRACCIÓN, TRANSFORMACIÓN Y CARGA (ETL):** Facilitan la recopilación de datos, la transformación de estos siendo más coherentes y su carga en un almacén de datos.

- Apache nifi
- Talend
- Microsoft SSIS.



- **ALMACENES DE DATOS (Data Warehouses):** Plataformas diseñadas para el análisis de datos:

- Amazon Redshift
- Google BigQuery
- Snowflake.



- **LENGUAJES DE PROGRAMACIÓN para ANÁLISIS DE DATOS:** Python con bibliotecas como: Pandas - NumPy - R - SQL, son utilizados para analizar y procesar datos.



- **Herramientas de VISUALIZACIÓN DE DATOS:** Permiten crear representaciones visuales comprensibles de datos para facilitar la toma de decisiones.

- Tableau
- Power BI
- Google Data Studio



+ a b | e a u



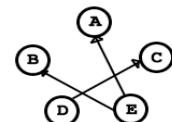
Formación de Datos Significativos:

La generación y manipulación de datos son procesos fundamentales respaldados por varias herramientas. Formar datos significativos implica entender la fuente de los datos, aplicar técnicas de manipulación y análisis, y utilizar herramientas especializadas para convertir los datos en información valiosa que respalde las operaciones empresariales.

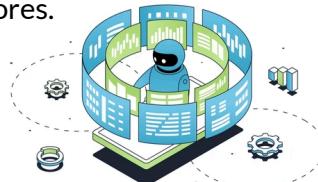
- **Análisis Descriptivo:** Explorar datos para entender su estructura y sus patrones. Herramientas de análisis descriptivo proporcionan una visión general de los datos.



- **Análisis Predictivo:** Utilizar algoritmos y modelos para prever ventas futuras basadas en datos históricos. Esto es crucial para la planificación de una estrategia.
- **Segmentación y Personalización:** Dividir datos en segmentos/clasificaciones para personalizar estrategias. Cómo segmentar clientes según su comportamiento de compra. O clasificar productos según su beneficio.



- **Automatización de Procesos:** Implementar la automatización de datos para mejorar la eficiencia operativa y reducir errores.



- **Monitoreo:** Utilizar herramientas de monitoreo en tiempo real para reaccionar rápidamente a cambios y eventos importantes.



IMPLICACIONES DEL USO DE DATOS

El uso de datos e información para respaldar los procesos de negocio implica una serie de implicaciones y consideraciones importantes:



Toma de decisiones basada en datos

Utilizar **datos e información** para **comunicar decisiones** empresariales permite una **toma de decisiones** más fundamentada y **objetiva**. Esto conduce a **decisiones más acertadas**, estás respaldadas por **evidencias y análisis** concretos.

Mejora de la eficiencia operativa

El **análisis de datos** puede **identificar** áreas de **ineficiencia** en los procesos empresariales y **proporcionar información** sobre cómo **mejorar la eficiencia operativa**. Esto incluiría:

- La identificación de cuellos de botella ⇒ [Áreas en un sistema donde el rendimiento está limitado por un componente que no mantiene el ritmo]
- La optimización de la cadena de suministro
- La reducción de costos operativos.

Personalización y mejor experiencia del cliente

Al **recopilar y analizar datos** sobre el **comportamiento** y las **preferencias** del **cliente**, las empresas pueden personalizar sus productos y servicios para satisfacer mejor las necesidades de los clientes. Esto conduce a una mayor satisfacción del cliente y a relaciones más sólidas con la marca.

Identificación de nuevas oportunidades de mercado

El **análisis de datos** puede ayudar a las empresas a **identificar nuevas oportunidades** de mercado y a **anticipar las tendencias** emergentes. Al comprender las necesidades y deseos del mercado, las empresas desarrollan estrategias de productos y marketing más efectivas.

Gestión del riesgo y cumplimiento normativo

El **uso de datos e información** puede ayudar a las empresas a **identificar y mitigar riesgos**, así como **garantizar** el cumplimiento de las **regulaciones y normativas**. Esto incluye la *detección de fraudes, la gestión de riesgos financieros y la protección de los datos*.



Por otro lado, su uso para respaldar los procesos empresariales es muy significativo y se manifiesta en varios aspectos:

La diferencia entre procesos empresariales y procesos de negocio es que el Proceso de Negocio se centra en actividades específicas que se realizan para lograr un objetivo dentro de una organización. Pero el Proceso de Empresa se refiere al conjunto más amplio de actividades y operaciones que constan del funcionamiento de la empresa. Los procesos de negocio son componentes importantes del proceso de empresa y contribuyen al logro de los objetivos y la misión de la organización en su conjunto.

Competitividad mejorada

Las empresas que utilizan datos efectivos obtienen una ventaja competitiva al tomar decisiones rápidas y precisas, adaptarse muy rápido a los cambios del mercado y ofrecer productos y servicios más relevantes a sus clientes.

Innovación y crecimiento

El acceso a datos y análisis sofisticados fomenta la innovación en las empresas, ya que proporciona información valiosa sobre nuevas oportunidades de mercado, tendencias emergentes y áreas de mejora operativa, impulsando el crecimiento empresarial a largo plazo.

Mejora de la rentabilidad

El uso eficaz de los datos mejora la rentabilidad, optimizando los procesos empresariales, reduciendo los costos operativos y aumentando los ingresos, por una mejor comprensión del mercado y una mayor satisfacción del cliente.

Gestión más efectiva de los recursos

La información derivada del análisis de datos ayuda a las empresas a asignar eficientemente sus recursos, como su capital humano, sus activos y su tiempo, logrando así sus objetivos comerciales de manera más efectiva.



El uso de datos e información en los procesos empresariales pueden tener un impacto profundo y positivo en la eficiencia, competitividad y éxito de una empresa. Sin embargo, es importante que las empresas aborden desafíos relacionados con la **privacidad de los datos**, la **seguridad cibernética** y la **calidad de los datos** para maximizar los beneficios del uso de datos y minimizar los riesgos potenciales.

TIPOS DE IMPLICACIONES

El uso de datos e información para respaldar los procesos empresariales conlleva importantes implicaciones **sociales**, **legales** y **éticas** que deben ser abordadas adecuadamente:

- Implicaciones Sociales:
 - + **Positivas:** El acceso a datos mejora la personalización de productos y servicios, ofreciendo nuevas experiencias. Aunque, la recopilación excesiva de datos genera preocupaciones de privacidad y autonomía.
 - **Negativas:** El uso indebido de datos puede resultar en discriminación o manipulación. Creando un modelo automatizado de procesos que provoque la pérdida de empleo.
 - Implicaciones Legales:
 - + **Positivas:** Las leyes de privacidad y protección de datos garantizan derechos individuales de cada persona y establecen ciertos estándares éticos. El acuerdo legal protege tanto a consumidores como a empresas.
 - **Negativas:** Si incumples las regulaciones establecidas, las sanciones serán severas. La violación de leyes de privacidad pueden generar desafíos legales.
 - Implicaciones Éticas:
 - + **Positivas:** La gestión ética de datos, sin malicia, mejora la confianza del consumidor y la reputación de la empresa. Los proyectos transparentes contribuyen a un desarrollo sostenible.
 - **Negativas:** Las decisiones con algoritmos pueden atraer curvas. Sin la transparencia al generar y usar datos, plantean preocupaciones éticas.
-

AMENAZAS COMUNES Y MEDIDAS DE MITIGACIÓN



Amenazas:

- 1) **Violación de Datos:** Accesos NO autorizados pueden llegar a resultar en la exposición de información importante.
- 2) **Ransomware** ⇒ Ataques de software malicioso que cifran datos y exigen un rescate.
- 3) **Phishing:** Engaño para obtener información confidencial con correos electrónicos o sitios web falsos.

Mitigación de Amenazas:

- **Contraseñas fuertes:** Implementar contraseñas robustas y cambiarlas regularmente.
- **Actualizaciones de Software:** Mantener sistemas y aplicaciones actualizados para corregir debilidades.
- **Concientización y Capacitación:** Enseñar a empleados y usuarios sobre técnicas seguras.
- **Cifrado de Datos:** Cifrar datos para proteger la confidencialidad de la información.
- **Respaldos Regulares:** Realizar copias de seguridad periódicas para disminuir la pérdida de datos.

El uso de datos e información en procesos empresariales plantea desafíos sociales, legales y éticos. La gestión transparente y ética, junto con medidas de seguridad sólidas, son muy importantes para abordar estas implicaciones y así, construir un entorno empresarial confiable, sostenible y exitoso.

IMPACTO DEL USO DE DATOS E INFORMACIÓN

Ahora, con todo lo visto anteriormente veremos el impacto que produce el uso de datos en cada implementación:

Toma de Decisiones Informada

La disponibilidad y análisis de datos ayuda a una toma de decisiones más informada y rápida, a nivel operativo y estratégico.

- ❖ **Ejemplo:** Una empresa minorista como Walmart puede ajustar sus estrategias de inventario basándose en datos de demanda diaria.



Eficiencia Operativa

La automatización respaldada por datos mejora la eficiencia operativa al optimizar procesos, reducir costos y minimizar errores.

- ❖ **EJ:** Sistemas de gestión de cadena de suministro que utilizan datos para prever demandas y optimizar la logística. Esta sistema es utilizado por Amazon:
Amazon tiene una eficiente cadena de suministro, con capacidad para entregar productos rápidamente y confiables a sus clientes en todo el mundo. Utiliza datos de diversas fuentes, como históricos de ventas, tendencias del mercado, patrones de compra y datos meteorológicos, para prever la demanda de productos con precisión.



Experiencia del Cliente

El análisis de datos permite enriquecer la experiencia del cliente, mejorando su satisfacción y fidelidad.

- ❖ **Ejemplo:** Plataformas de streaming, como Netflix, HBO max, que recomiendan contenido basado en elecciones anteriores de los usuarios.



Innovación y Adaptabilidad

El uso creativo de datos genera innovación y adaptación a cambios en el mercado.

- ❖ **Ejemplo:** Airbnb recopila y analiza una gran cantidad de datos sobre viajes y preferencias de los usuarios para adaptarse constantemente a las demandas del mercado. Así, utiliza los datos para mejorar la experiencia de usuario, desarrollar nuevos servicios y fijar precios y disponer alojamientos.



Gestión de Riesgos

Con el monitoreo constante de datos se mejora la gestión de riesgos, ayudando a prevenir y mitigar amenazas potenciales.

- ❖ **Ejemplo:** Empresas tecnológicas como Google y Microsoft utilizan análisis de riesgos para garantizar la seguridad de sus plataformas en línea, proteger los datos de los usuarios y prevenir ciberataques.



Competitividad y Posicionamiento en el Mercado

Anticipar y adaptarse rápidamente a tendencias del mercado proporciona una ventaja competitiva.

- ❖ **Ejemplo:** Nike es conocida por su capacidad para anticipar y acumular las tendencias en la industria del deporte y la moda, innovando en su diseño y tecnología, manteniendo su envergadura y liderazgo en el mercado del calzado deportivo.



Cumplimiento Normativo

La gestión adecuada de datos debe ser completada con el cumplimiento normativo, evitando sanciones y protegiendo su reputación.

- ❖ **Ejemplo:** Empresas de salud, como *McKesson Corporation* que se adhiere a estrictos estándares de cumplimiento normativo y trabaja para proteger la confidencialidad y la integridad de los datos de salud de sus clientes.

Desafíos Éticos y Sociales



Aunque el uso de datos ofrece muchos beneficios, como ya hemos visto, también plantea desafíos éticos y sociales relacionados con la privacidad y los prejuicios.

- ❖ **Ejemplo:** Mark Zuckerberg, CEO de Facebook, se enfrentó a críticas y a un juicio en 2018, donde se reveló que datos de millones de usuarios fueron utilizados sin su consentimiento para fines políticos. Esto resalta los desafíos éticos y de privacidad asociados con el uso de datos en plataformas como Facebook.



Personalización y Segmentación de Mercado

El análisis de datos permite a las empresas personalizar sus productos y clasificar mensajes a segmentos específicos de clientes.

- ❖ **Ejemplo:** Como las estrategias de marketing que utilizan datos demográficos y de comportamiento para crear anuncios.

El impacto del uso de datos e información en los procesos empresariales es profundamente positivo, impulsando la eficiencia, la innovación y la competitividad. Sin embargo, es crucial abordar desafíos éticos y sociales para garantizar un uso responsable y sostenible de la información.

TECNOLOGÍAS

La ciencia de datos es muy importante para los procesos de negocio, ofreciendo herramientas y tecnologías que permiten extraer conocimientos valiosos a partir de grandes conjuntos de datos. Estas tecnologías no solo optimizan la eficiencia operativa, sino que también informan decisiones estratégicas.

1. Análisis Predictivo:

- Python con bibliotecas como *Scikit-Learn*, *TensorFlow* para aprendizaje automático.
 - > Prevé tendencias futuras, mejora la planificación y anticipa demandas del mercado.

2. Visualización de Datos:

- *Tableau, Power BI, D3.js.*
- Facilita la comprensión de datos complejos, ayuda en la presentación de informes y apoya la toma de decisiones basada en datos.

3. Sistemas de Gestión de Bases de Datos:

- *MySQL, PostgreSQL, MongoDB.*
- Garantiza un almacenamiento eficiente y seguro de datos, facilitando la recuperación rápida y precisa cuando sea necesario.

4. Big Data y Tecnologías de Almacenamiento:

- *Hadoop, Apache Spark, Amazon S3.*
- Maneja grandes volúmenes de datos, permitiendo un análisis más completo y detallado.

De estas tecnologías ya hemos hablado anteriormente y ya tenemos claro que son y qué hacen.
Aun así, hay mas:

5. Minería de Datos:

- *RapidMiner, KNIME, Weka.*
- Identifica patrones y relaciones en conjuntos de datos, revelando información valiosa para la toma de decisiones.

6. Inteligencia Artificial (IA) y Aprendizaje Automático (ML):

- *TensorFlow, PyTorch, Microsoft Azure ML.*
- Automatiza procesos, mejora la personalización de productos y servicios, y optimiza la toma de decisiones.

7. Procesamiento de Lenguaje Natural (NLP):

- *NLTK (Natural Language Toolkit), spaCy.*
- Permite la comprensión y análisis de datos no estructurados, como comentarios de clientes o redes sociales.

8. Automatización de Procesos Robóticos (RPA):

- *UiPath, Automation Anywhere.*
- Mejora la eficiencia mediante la automatización de tareas repetitivas, liberando tiempo para actividades más estratégicas.

9. Plataformas de Nube:

- AWS, Azure, Google Cloud.
- Ofrece escalabilidad, acceso remoto y flexibilidad para implementar y gestionar herramientas de ciencia de datos.

10. Realidad Virtual (VR) y Realidad Aumentada (AR):

- VR: *Google Cardboard, HTC Vive, Meta Quest.* AR: *Microsoft HoloLens, Vuzix Blade, Epson Moverio.*
- Crear experiencias inmersivas que mejoran la interacción con los usuarios y ofrecen nuevas formas de comunicación.

11. Blockchain:

- *Hyperledger, Bitcoin, Corda*
- Almacena y transmite la información de manera segura, transparente y descentralizada. Garantiza que la información sea inmutable y no pueda ser alterada

12. Internet de las cosas (IoT):

- ThingWorx, Google Cloud IoT, Siemens MindSphere

- Recopila y transmite información en tiempo real, automatizando los procesos y la creación de nuevos modelos de negocio

Con estas tecnologías se trabaja la ciencia de datos, transformando datos en información accionable. Su integración en los procesos de negocio optimiza la eficiencia operativa y también permite a las organizaciones tomar decisiones más informadas y estratégicas.

TECNOLOGÍAS PARA APOYAR LOS PROCESOS DE NEGOCIO E INFORMAR LAS DECISIONES

La ciencia de datos utiliza diferentes tipos de técnicas y metodologías para analizar datos, extraer información importante y obtener ***insights*** que pueden ser utilizados para optimizar operaciones, identificar oportunidades de crecimiento, mitigar riesgos y mejorar la eficiencia.

insights: descubrimientos significativos que proporcionan una comprensión más clara de una situación, problema o fenómeno

En cuanto a la organización de los datos, hay que comprender cómo se clasifican:
ESTRUCTURADOS, SEMIESTRUCTURADOS o NO ESTRUCTURADOS:

TIPOS	DEFINICIÓN	EJEMPLOS
Datos ESTRUCTURADOS	Son datos organizados en un formato predefinido y fácilmente identificable. Normalmente, se almacenan en bases de datos relacionales o en hojas de cálculo. Los datos tienen un esquema claro y definido, facilitando su procesamiento y análisis.	<ul style="list-style-type: none"> - Información de bases de datos de clientes - Registros de transacciones financieras - Datos de inventario.
Datos SEMIESTRUCTURADOS	Son datos que no tienen un formato predefinido y no se ajustan a las estructuras de datos como las bases de datos relacionales. Así mismo se organizan, como <i>etiquetas</i> o <i>metadatos</i> que permiten su clasificación y búsqueda.	<ul style="list-style-type: none"> - Documentos XML - Archivos JSON - Correos electrónicos - Archivos de registro.
Datos NO ESTRUCTURADOS	Son datos que no tienen un formato predefinido ni una estructura clara. Son la	<ul style="list-style-type: none"> - Texto libre - Imágenes - Audio

	<p>mayor parte de los datos actuales, siendo difíciles de procesar y de analizar utilizando métodos tradicionales.</p>	<ul style="list-style-type: none"> - Video - Redes sociales - Páginas web
--	--	--

Las herramientas y tecnologías de ciencia de datos están diseñadas para manejar y analizar datos en todos estos formatos.

Los datos ESTRUCTURADOS, que se organizan en tablas con filas y columnas definidas, son manejados por lenguajes de consulta como *SQL*.

Además, herramientas de programación como *R* y *Python* con sus librerías *pandas* y *NumPy* son muy utilizadas para manipular y analizar datos ESTRUCTURADOS. Estas librerías ofrecen funciones y métodos para realizar operaciones de limpieza, transformación y visualización de datos.

Los datos SEMIESTRUCTURADOS, como archivos *JSON* o *XML*, pueden ser procesados y analizados con lenguajes de programación como *Python*, aprovechando sus capacidades para trabajar con datos *semi-organizados* y extraer información importante de ellos.

Para abordar datos NO ESTRUCTURADOS, como texto libre, imágenes y videos, se recurre a técnicas de procesamiento de datos, como técnicas de visión por computadora y aprendizaje profundo, para extraer características y patrones relevantes. Bibliotecas como *OpenCV* y frameworks de aprendizaje como *TensorFlow* y *PyTorch* son fundamentales.

R, *SQL*, *Python* con sus librerías *pandas* y *NumPy*, así como herramientas como Apache Hadoop, Spark, y las bases de datos NoSQL, ofrecen capacidades para procesar grandes volúmenes de datos en diferentes formatos y realizar análisis complejos.

Al organizar y analizar datos estructurados, semiestructurados y no estructurados, las herramientas y tecnologías de ciencia de datos ayudan a las empresas a comprender mejor su entorno, tomar decisiones informadas y obtener ventajas competitivas en el mercado.

BENEFICIOS DE UTILIZAR LA CIENCIA DE DATOS PARA RESOLVER PROBLEMAS

La ciencia de datos ofrece una amplia gama de beneficios para resolver problemas en escenarios reales. Algunos de estos beneficios son convenientes al utilizar datos estructurados, semiestructurados o no estructurados:

Beneficios de utilizar la ciencia de datos:

La ciencia de datos permite identificar patrones, tendencias y relaciones ocultas, lo que ayuda a las organizaciones a **tomar decisiones** basadas en evidencias y datos concretos.

Al analizar datos históricos y en tiempo real, las empresas pueden identificar áreas de mejora en sus **procesos operativos, logísticos y de producción**, lo que les permite optimizar la eficiencia y reducir costos.

Mediante el análisis de datos del cliente, las empresas pueden entender mejor las **preferencias y comportamientos de sus clientes**, lo que les permite ofrecer productos y servicios personalizado.

Las técnicas de ciencia de datos, como el análisis predictivo y la minería de datos, pueden ser utilizadas para identificar patrones de comportamiento fraudulentos o de alto riesgo en transacciones financieras, seguros, salud, entre otros, ayudando a **prevenir pérdidas y mitigar riesgos**.

Al analizar datos de retroalimentación de los clientes, datos de sensores y otros tipos, las empresas pueden **identificar áreas de mejora en sus productos y servicios**, así como **oportunidades para la innovación y el desarrollo de nuevos productos**.

Ejemplos de situaciones en las que sería conveniente utilizar Datos ESTRUCTURADOS, SEMIESTRUCTURADOS o NO ESTRUCTURADOS:

➤ **Datos ESTRUCTURADOS:**

El estanco desea entender ciertos patrones de compra de sus clientes para personalizar ofertas y mejorar la gestión de inventario.

El estanco recopila datos estructurados de transacciones de venta que incluyen:

Productos:

- Nombre del producto
- Cantidad
- Precio
- Fecha y hora de la compra
- Total de venta por producto

Clientes:

- ID del cliente
- Nombre del cliente
- Número de teléfono
- Dirección

Método de Pago:

- Efectivo, tarjeta de crédito, tarjeta de débito, etc.

Datos de Inventario:

- Cantidad disponible de cada producto.

Utilizando estos datos, el estanco puede llevar a cabo análisis de patrones de compra para identificar tendencias y preferencias de sus clientes.

➤ **Datos SEMIESTRUCTURADOS:**

Registro de reseñas de libros realizadas por los clientes de una librería . Se podría estructurar:

Registro de Reseñas:

- Título del libro
- Nombre del cliente
- Fecha
- Calificación

- Reseña
- Identificación del libro (ISBN u otro)
- correo electrónico o numero de teléfono
- Información adicional sobre el cliente

Estos son datos semiestructurados, tienen una estructura básica predefinida, pero el texto de la reseña puede variar en longitud, contenido y formato, lo que no siempre sigue una estructura rígida.

El registro de reseñas de libros podría ser almacenado en una base de datos NoSQL, como *MongoDB*, que permite la flexibilidad necesaria para manejar datos semiestructurados y no estructurados eficientemente.

Utilizando estos datos, la librería podría realizar acciones como:

- Mostrar reseñas y opiniones en su web o aplicación para ayudar a clientes a tomar decisiones sobre qué libros leer, mejorando la experiencia del usuario.
- Analizar si las reseñas son positivas, neutrales o negativas, utilizando técnicas de procesamiento del lenguaje natural (*NLP*) para comprender la opinión de los clientes sobre un libro en particular y extraer la información relevante.
- Identificar tendencias y patrones en las reseñas, entendiendo qué libros son más apreciados por los clientes y cuáles podrían necesitar mejorar.

➤ **Datos NO ESTRUCTURADOS:**

Al comparar los detalles de un crimen reciente con crímenes similares ocurridos en el pasado, los investigadores podrían identificar similitudes en el modus operandi, los perfiles de los sospechosos o las circunstancias que rodearon los crímenes, ayudando en la identificación de patrones y en la resolución del caso.

El análisis de informes policiales, testimonios de testigos, grabaciones de llamadas de emergencia y artículos de noticias sobre un crimen específico y otro similar ocurrido en el pasado.

Además, el análisis de la reacción del público en las redes sociales y en los medios de comunicación puede proporcionar información sobre el impacto del crimen en la comunidad, así como posibles testigos o personas involucradas que podrían no haber sido identificadas.

Informes Policiales y Testimonios de Testigos:

- Descripción del crimen, incluyendo fecha y hora.
- Testimonios o personas involucradas.
- Descripción de sospechoso y víctimas.
- modus operandi del delincuente.
- Escena del crimen.

Grabaciones de Llamadas de Emergencia:

- Transcripciones de llamadas de emergencias relacionadas con el crimen.
- Naturaleza y la gravedad del crimen.
- Detalles proporcionados por testigos o víctimas durante la llamada.

Publicaciones en Redes Sociales:

- Comentarios, fotos o videos relacionados con el crimen compartidos en plataformas como Twitter, Facebook, Instagram, etc.
- Opiniones y reacciones de la comunidad en línea sobre el caso.

- Información adicional sobre personas y eventos relacionados con el crimen.

Artículos de Noticias:

- Reportajes de medios de comunicación sobre el crimen.
- Análisis y comentarios de expertos sobre el caso.
- Detalles históricos sobre crímenes similares del pasado con características similares.

Estos datos no estructurados pueden ser analizados utilizando técnicas de procesamiento del lenguaje natural (NLP) para extraer información relevante, identificar patrones y relaciones, y comprender mejor la naturaleza del crimen y su contexto en cuanto al pasado.

USO DE TÉCNICAS

Para diseñar cualquier solución de ciencia de datos implica seguir un proceso sistemático que incluya la identificación del problema, la recopilación y exploración de datos, el modelado y análisis, y la implementación de la solución. Índice:

1. Identificación del Problema
2. Recopilación y Preparación de Datos
3. Exploración de Datos
4. Modelado y Análisis
5. Interpretación de Resultados
6. Implementación y Monitoreo
7. Evaluación y Mejora Continua

Al seguir estos pasos, podremos diseñar una solución de ciencia de datos efectiva que apoye la toma de decisiones sistemáticamente y basada en datos. Es importante involucrar a las partes interesadas relevantes en cada etapa del proceso para garantizar que la solución sea relevante y útil para la organización.

Problema Empresarial = Mejora de la Experiencia del Usuario en una Biblioteca que alquila Libros

1) PROBLEMA:

Una empresa que ofrece servicios de alquiler de libros en su biblioteca desea mejorar la experiencia del usuario durante todo el proceso de alquiler. El objetivo es aumentar la retención de usuarios, con nuevas estrategias para mantener a los lectores comprometidos y activos, maximizando su satisfacción y facilitando el acceso a los libros de su interés.

2) OBJETIVO:

Mejorar la experiencia del usuario en el proceso de alquiler de libros dentro de la plataforma de la biblioteca. Nos centraremos en la experiencia de usuario al añadir un libro. Después podríamos crear un apartado para usuario y otro para transacciones. Pero en esta parte solo nos centraremos en añadir y ver los libros disponibles.

3) REQUISITOS y SOLUCIÓN:

Algunos requisitos que debemos realizar para quitarnos esos problemas, y que estrategias podemos usar para mitigarlos:

- **Análisis de la experiencia del usuario:** Se debe analizar la experiencia del usuario en cada etapa del proceso de alquiler, desde la búsqueda de libros hasta la devolución.
 - **Solución:**
Realizar un análisis exhaustivo de la experiencia del usuario en cada etapa del proceso de alquiler, identificando áreas de mejora y oportunidades de optimización.
- **Identificar los puntos flojos:** Es necesario identificar qué puntos son débiles y qué áreas de mejora encontramos en el proceso de alquiler, como la navegación, la disponibilidad de libros, el proceso de reserva y devolución, etc.
 - **Solución:**
Identificar las desventajas y las barreras que dificultan la experiencia de usuario, como problemas de búsqueda, lentitud en la navegación, procesos de reserva complicados, etc. Con distintas formas:
 - Con herramientas de análisis de datos para rastrear el comportamiento de los usuarios.
 - Patrones de comportamiento con dificultades para los usuarios, como clics repetidos o búsquedas frustradas.
 - Pedir opiniones de los usuarios frecuentes.
 - Realiza pruebas de usuario
 - Solicita opiniones y comentarios de los empleados
 - Comparar con otras plataformas de alquiler de libros
- **Optimización de la interfaz de usuario:** Se necesita mejorar la interfaz de usuario para que sea intuitiva, fácil de navegar y accesible para todos los usuarios.
 - **Solución:**
Esto se puede conseguir, simplificando del catálogo, agregar otras funciones de búsqueda, y optimizar el proceso de reserva y devolución de libros.
- **Implementación de Retroalimentación:** Se deben implementar mecanismos para la retroalimentación de los usuarios y utilizarla para mejorar la experiencia del usuario.
 - **Solución:**
Para implementar sistemas de retroalimentación, se hará a través de encuestas, comentarios o análisis de datos de uso, y utilizar esta información para realizar mejoras continuas en la experiencia del usuario.
- **Personalización de Servicios:** Podemos ofrecer otros servicios según las preferencias y necesidades de los usuarios, como sugerencias de libros, notificaciones de disponibilidad, etc. (OPCIONAL)

4) BENEFICIOS ESPERADOS:

- Mejora significativa en la experiencia del usuario durante el proceso de alquiler.
- Aumento en la retención de usuarios al proporcionar una experiencia más satisfactoria y fácil.
- Incremento en el número de alquileres de libros.

- Recopilación de datos valiosos sobre las preferencias y necesidades de los usuarios para informar decisiones empresariales futuras.

Para implementar una solución de ciencia de datos para apoyar la toma de decisiones relacionadas con el problema, se debe seguir un proceso:

- 1** Preparación de datos
- 2** Construcción de datos y Evaluación de modelos,
- 3** La implementación de la solución.
- 4** Monitoreo

Implementar una solución de ciencia de datos:

Paso 1: Preparación de Datos:

- Reunir los datos importantes para el problema. de la bases de datos
- Eliminar valores atípicos, datos redundantes o irrelevantes, datos faltantes y errores.
- Exploración de Datos: Realizar análisis para comprender la estructura de los datos, identificar patrones y relaciones, y determinar la distribución de las variables.

Paso 2: Construcción y Evaluación de Modelos:

- Elegir el Modelo de ciencia de datos adecuado, basado en la naturaleza del problema y los datos disponibles.
- Dividir tus datos en conjuntos de entrenamiento y prueba. Luego, entrenar los modelos utilizando el conjunto de entrenamiento.
- Evaluación de rendimiento de los modelos con un conjunto de pruebas.

Paso 3: Implementación de la Solución:

- Implementar la solución de ciencia de datos utilizando el modelo seleccionado.
- Integración de la Solución: Integrar una solución y asegurarse de que los usuarios finales puedan acceder y utilizarla.
- Realizar pruebas exhaustivas para garantizar que la solución funcione correctamente y cumpla con los requisitos establecidos.

Paso 4: Monitoreo y Mantenimiento:

- Implementar sistema de monitoreo para supervisar el rendimiento de la solución en tiempo real. Con resolución de problemas.
- Realizar actualizaciones y mejoras regulares.
- Proporcionar soporte a los usuarios para garantizar que puedan utilizar la solución de manera efectiva y comprender sus resultados.

SOFTWARE

DISEÑO DE SOLUCIÓN

Para el proceso empresarial de alquiler de libros, es fundamental contar con una solución de ciencia de datos que permita recopilar, analizar y utilizar los datos de manera efectiva para mejorar la experiencia del usuario y optimizar las operaciones.

Nos apoyarnos en un Sistema de Gestión de Base de Datos (DBMS) que organice los datos estructuradamente, ya que esto facilitará su almacenamiento, consulta y análisis.

DISEÑO CONCEPTUAL

En el diseño conceptual, identificamos las entidades principales y las relaciones entre ellas.

Entidades:

- 1) **Libro:** Contiene información sobre los libros disponibles para alquilar.

ATRIBUTOS: *ID_Libro*, *título*, *autor*, *género*, *editorial*, *año de publicación* y si está *Disponible* o no.

- 2) **Usuario:** Representa a los usuarios que utilizan el servicio de alquiler de libros.

ATRIBUTOS: *ID_Usuario*, *nombre*, *apellidos*, *dirección* y *correo electrónico*.

- 3) **Transacción:** Registra las transacciones de alquiler de libros.

ATRIBUTOS: *ID_Transaccion*, *fecha de alquiler*, *fecha de devolución*, *estado del libro* (*alquilado/devuelto*).

Relaciones:

- **USUARIO - LIBRO:**

- Un usuario puede alquilar múltiples libros.
- Un libro puede ser alquilado por varios usuarios en diferentes momentos.
- Es decir, es una relación de muchos a muchos (N:M).

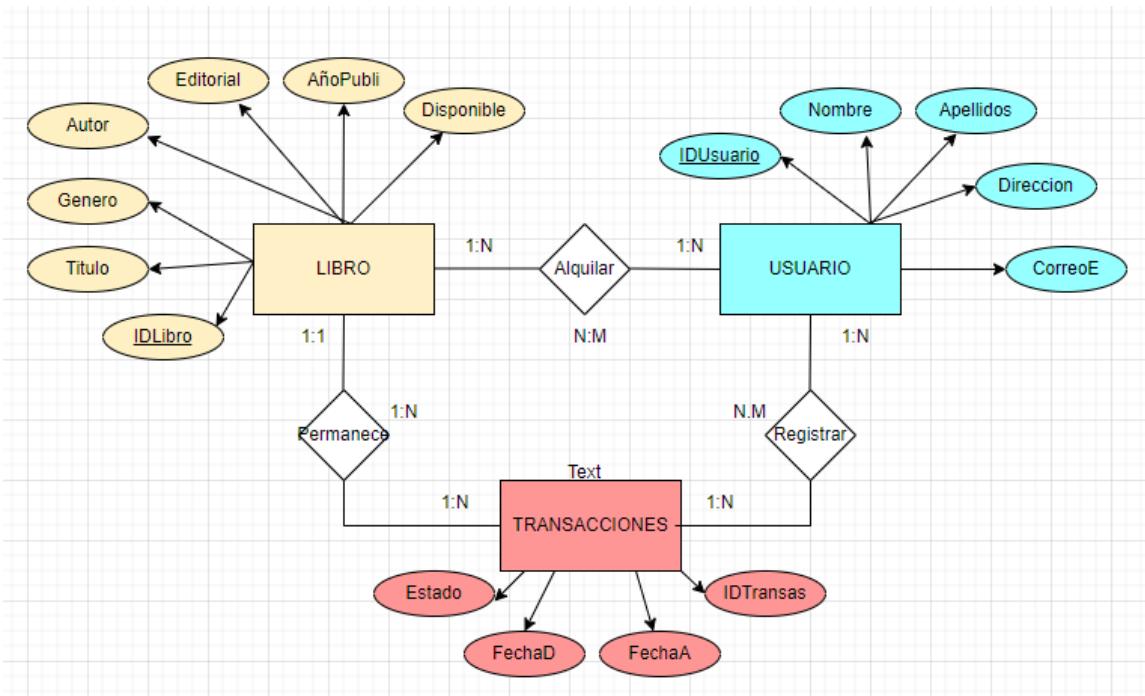
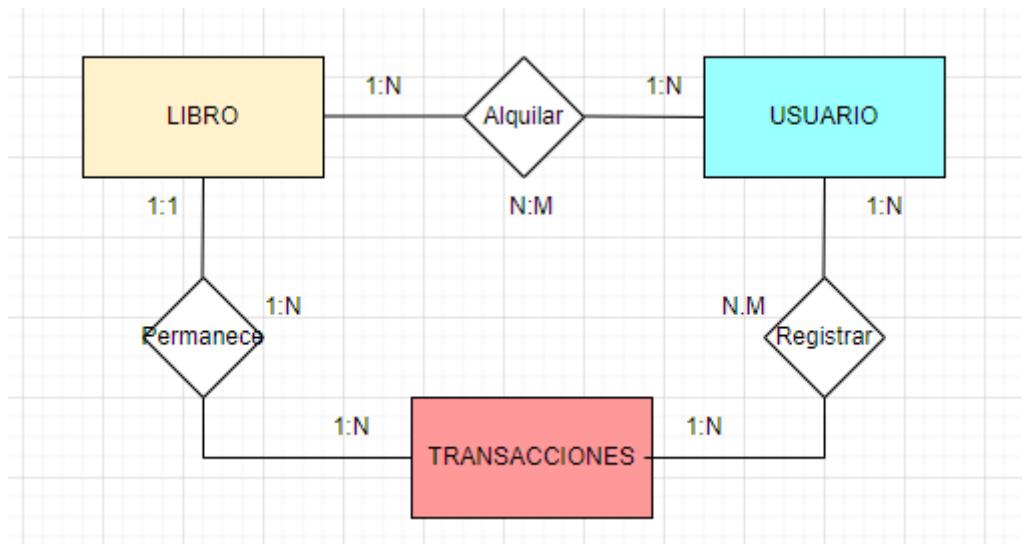
- **TRANSACCIÓN - LIBRO:**

- Puede haber varias transacciones del mismo libro.
- Pero un libro solo tendrá una transacción en un momento dado.
- Esto indica una relación de uno a muchos (1:N), ya que un libro tiene múltiples transacciones pero cada transacción sólo está asociada a un libro.

- **TRANSACCIÓN - USUARIO:**

- Un usuario puede tener varias transacciones.
- Una transacción puede involucrar varios usuarios, como un libro puede ser alquilado por más de un usuario a la vez.
- Esto es nuevamente una relación de muchos a muchos (N:M).

DIAGRAMA DE FLUJO



Al tener dos relaciones con muchos con muchos(N:M) debemos crear otras dos tablas y con las claves de las tablas en relación.

DISEÑO LÓGICO

En el diseño lógico, convertimos el diseño conceptual en un modelo relacional utilizando tablas y relaciones.

1) Tabla Libro:

Columnas: ID_Libro (PK), Titulo, Autor, Genero, Editorial, Año_Publicacion, Disponible

2) Tabla Usuario:

Columnas: *ID_Usuario* (PK), *Nombre*, *Apellidos*, *Direccion*, *Correo*

3) Tabla Transaccion:

Columnas: *ID_Transaccion*, *ID_Usuario* (FK), *ID_Libro* (FK), *Fecha_Alquiler*,
Fecha_Devolucion, *Estado*

- **Claves Foraneas:**

- *ID_Usuario* es una FK de la Tabla Usuario.
- *ID_Libro* es una FK de la Tabla Libro.

4) Tabla Alquilar:

- Esta tabla registraría los préstamos de libros por parte de los usuarios.

- **Columnas:** *ID_Alquilado* (PK), *ID_Usuario* (FK), *ID_Libro* (FK)

- **Claves Foraneas:**

- *ID_Usuario* es una FK de la tabla Usuario.
- *ID_Libro* es un FK de la tabla Libro.

5) Tabla Registrar:

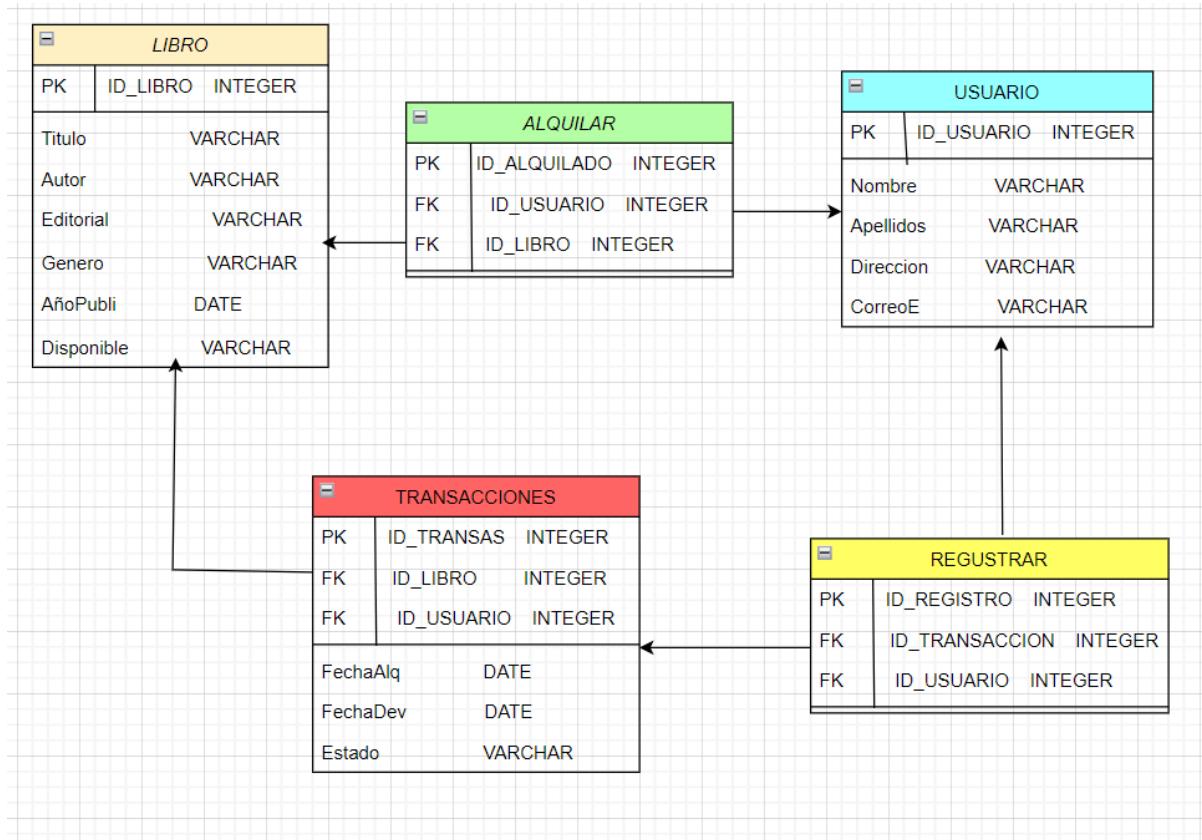
- Esta tabla registra las transacciones asociadas con los usuarios.

- **Columnas:** *ID_Registro* (PK), *ID_Transaccion* (FK), *ID_Usuario* (FK)

- **Claves Foraneas:**

- *ID_Transaccion* es un FK de la tabla Transaccion.
- *ID_Usuario* es un FK de la tabla Usuario.

DIAGRAMA UML



DISEÑO FÍSICO

En el diseño físico, consideramos aspectos de implementación y optimización en el DBMS, como la creación de índices, particiones, ajustes de rendimiento, etc. Para mejorar el rendimiento y la eficiencia en el manejo de los datos.

Implementación en un DBMS:

1. Elegir un DBMS relacional como MySQL, PostgreSQL, etc.
2. Crearemos índices en campos clave para mejorar la velocidad de búsqueda.
3. Configuraremos copias de seguridad periódicas para garantizar la integridad de los datos.
4. Implementaremos funciones para operaciones complejas.

Con este diseño de solución de ciencia de datos, la empresa de alquiler de libros puede gestionar eficazmente su base de datos y utilizarla para optimizar sus operaciones y conseguir nuestros objetivos de mejorar la experiencia del usuario. Es importante mantener el diseño actualizado y ajustarlo según las necesidades cambiantes del negocio y las demandas del mercado.

Con esto claro vamos ha empezar:

1. Elegiremos como DBMS relacional **PostgreSQL**, ya que lo hemos utilizado en otra ocasión, y ya estamos familiarizados con su funcionamiento.

Así que empezaremos a crear las tablas y añadir los datos, con ayuda del diseño lógico de la bbdd.

- 1) CREACIÓN DE TABLAS en PostgreSQL, que debido a ciertos errores se cambiaría a MySQL:

```
CREATE TABLE libro (
    ID_libro SERIAL PRIMARY KEY,
    Titulo VARCHAR(200) NOT NULL,
    Autor VARCHAR(100) NOT NULL,
    Editorial VARCHAR(100),
    Genero VARCHAR(200),
    AñoPubli      INT,
    Disponible BOOLEAN
);
```

```
CREATE TABLE usuario (
    ID_usuario SERIAL PRIMARY KEY,
    Nombre VARCHAR(200),
    Apellidos VARCHAR(200),
    Direccion VARCHAR(200),
    CorreoElec VARCHAR(200)
);
```

```
CREATE TABLE transacciones(
    ID_transaccion SERIAL PRIMARY KEY,
    ID_usuario INT,
    ID_libro INT,
    Fecha_Alquiler DATE,
    Fecha_Devolucion DATE,
    Estado VARCHAR(50),
    FOREIGN KEY (ID_usuario) REFERENCES usuario(ID_usuario),
    FOREIGN KEY (ID_libro) REFERENCES libro(ID_libro)
);
```

```
CREATE TABLE alquilar (
    ID_Alquilado INT PRIMARY KEY,
    ID_usuario INT,
    ID_libro INT,
    FOREIGN KEY (ID_usuario) REFERENCES usuario(ID_usuario),
```

```
FOREIGN KEY (ID_libro) REFERENCES libro(ID_libro)
);
```

```
CREATE TABLE registrar (
    ID_Registro INT PRIMARY KEY,
    ID_Transaccion INT,
    ID_Usuario INT,
    FOREIGN KEY (ID_transaccion) REFERENCES transacciones(ID_transaccion),
    FOREIGN KEY (ID_usuario) REFERENCES usuario(ID_usuario)
);
```

2) INSERTAR DATOS:

Ahora mismo lo que vamos ha hacer es añadir los datos de la **Tabla Libro**, ya que la propia aplicación nos ofrecerá la información de los libros disponibles. Y los datos de las demás tablas se irán añadiendo a medida que el usuario ingrese sus preferencias.

Para ello utilizamos el siguiente comando:

```
INSERT INTO libro (Titulo, Autor, Editorial, Genero, AñoPubli, Disponible)
VALUES ('titulo', 'Autor1', 'Editorial1', 'Genero1', año, true);
```

Como el ID se ingresa solo gracias al SERIAL, no hace falta añadirlo.

```
INSERT INTO libro (Titulo, Autor, Editorial, Genero, AñoPubli, Disponible)
VALUES ('El nombre del viento', 'Patrick Rothfuss', 'DAW Books', 'Fantasía', 2007,
true);
```

Insertamos unas cuantas así.

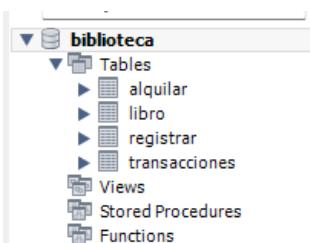
Comprobamos con el comando SELECT:

```
SELECT * FROM libro
```

id_libro [PK] integer	título character varying (200)	autor character varying (100)	editorial character varying (100)	genero character varying (200)	añopubli integer	disponible boolean
1	El nombre del viento	Patrick Rothfuss	DAW Books	Fantasía	2007	true
2	El temor de un hombre sabio	Patrick Rothfuss	DAW Books	Fantasía	2011	true
3	El Imperio Final	Brandon Sanderson	Tor Books	Fantasía	2006	true
4	El pozo de la ascensión	Brandon Sanderson	Tor Books	Fantasía	2007	true
5	El héroe de las eras	Brandon Sanderson	Tor Books	Fantasía	2008	true
6	Rebelión en la Bounty	John Boyne	C.S. Forester	Histórica	1932	true
7	Cien años de soledad	Gabriel García Márquez	Editorial Sudamericana	Realismo mágico	1967	true
8	El señor de los anillos: La comunidad del anillo	J.R.R. Tolkien	Allen & Unwin	Fantasía	1954	true
9	1984	George Orwell	Secker & Warburg	Ciencia ficción	1949	true
10	Matar a un ruiseñor	Harper Lee	J.B. Lippincott & Co.	Novela	1960	true
11	Harry Potter y la piedra filosofal	J.K. Rowling	Bloomsbury Publishing	Fantasía	1997	true
12	Orgullo y prejuicio	Jane Austen	T. Egerton, Whitehall	Novela romántica	1813	true
13	La sombra del viento	Carlos Ruiz Zafón	Planeta	Misterio	2001	true
14	Crónica de una muerte anunciada	Gabriel García Márquez	Editorial Sudamericana	Novela	1981	true
15	El alquimista	Paulo Coelho	HarperCollins	Ficción	1988	true
16	El principito	Antoine de Saint-Exupéry	Reynal & Hitchcock	Literatura infantil	1943	true
17	El código Da Vinci	Dan Brown	Doubleday	Ficción de misterio	2003	true
18	Crimen y castigo	Fyodor Dostoevsky	The Russian Messenger	Novela psicológica	1866	true
19	La ciudad y los perros	Mario Vargas Llosa	Seix Barral	Novela	1963	true
20	Don Quijote de la Mancha	Miguel de Cervantes	Juan de la Cuesta	Novela satírica	1605	true

Con MySQL sería prácticamente igual:

Crear bbdd + tablas + datos en libro + comprobación:



```
CREATE DATABASE Biblioteca;
USE Biblioteca;
```

```
CREATE TABLE libro (
    ID_libro INT AUTO_INCREMENT PRIMARY KEY,
    Titulo VARCHAR(200) NOT NULL,
    Autor VARCHAR(100) NOT NULL,
    Editorial VARCHAR(100),
    Genero VARCHAR(200),
    AñoPubli INT,
    Disponible BOOLEAN
);
```

```
select * from libro;
CREATE TABLE usuario (
    ID_usuario INT AUTO_INCREMENT PRIMARY KEY,
    Nombre VARCHAR(200),
    Apellidos VARCHAR(200),
    Direccion VARCHAR(200),
    CorreoElec VARCHAR(200)
);
drop table usuario;
```

```
CREATE TABLE transacciones (
    ID_transaccion INT AUTO_INCREMENT PRIMARY KEY,
```

```

ID_libro INT,
FOREIGN KEY (ID_libro) REFERENCES libro(ID_libro)
);

CREATE TABLE alquilar (
ID_Alquilado INT AUTO_INCREMENT PRIMARY KEY,
ID_libro INT,
FOREIGN KEY (ID_libro) REFERENCES libro(ID_libro)
);

CREATE TABLE registrar (
ID_Registro INT AUTO_INCREMENT PRIMARY KEY,
ID_Transaccion INT,
FOREIGN KEY (ID_Transaccion) REFERENCES transacciones(ID_transaccion)
);

-- Insertar los libros
INSERT INTO libro (Titulo, Autor, Editorial, Genero, AñoPubli, Disponible)
VALUES ('El nombre del viento', 'Patrick Rothfuss', 'DAW Books', 'Fantasía', 2007, true),
('El temor de un hombre sabio', 'Patrick Rothfuss', 'DAW Books', 'Fantasía', 2011, true),
('El Imperio Final', 'Brandon Sanderson', 'Tor Books', 'Fantasía', 2006, true),
('El pozo de la ascensión', 'Brandon Sanderson', 'Tor Books', 'Fantasía', 2007, true),
('El héroe de las eras', 'Brandon Sanderson', 'Tor Books', 'Fantasía', 2008, true),
('Rebelión en la Bounty', 'John Boyne', 'C.S. Forester', 'Histórica', 1932, true),
('Cien años de soledad', 'Gabriel García Márquez', 'Editorial Sudamericana', 'Realismo mágico', 1967, true),
('El señor de los anillos: La comunidad del anillo', 'J.R.R. Tolkien', 'Allen & Unwin', 'Fantasía', 1954, true),
('1984', 'George Orwell', 'Secker & Warburg', 'Ciencia ficción', 1949, true),
('Matar a un ruiseñor', 'Harper Lee', 'J.B. Lippincott & Co.', 'Novela', 1960, true),
('Harry Potter y la piedra filosofal', 'J.K. Rowling', 'Bloomsbury Publishing', 'Fantasía', 1997, true),
('Orgullo y prejuicio', 'Jane Austen', 'T. Egerton, Whitehall', 'Novela romántica', 1813, true),
('La sombra del viento', 'Carlos Ruiz Zafón', 'Planeta', 'Misterio', 2001, true),
('Crónica de una muerte anunciada', 'Gabriel García Márquez', 'Editorial Sudamericana', 'Novela', 1981, true),
('El alquimista', 'Paulo Coelho', 'HarperCollins', 'Ficción', 1988, true),
('El principito', 'Antoine de Saint-Exupéry', 'Reynal & Hitchcock', 'Literatura infantil', 1943, true),
('El código Da Vinci', 'Dan Brown', 'Doubleday', 'Ficción de misterio', 2003, true),
('Crimen y castigo', 'Fyodor Dostoevsky', 'The Russian Messenger', 'Novela psicológica', 1866, true),
('La ciudad y los perros', 'Mario Vargas Llosa', 'Seix Barral', 'Novela', 1963, true),
('Don Quijote de la Mancha', 'Miguel de Cervantes', 'Juan de la Cuesta', 'Novela satírica', 1605, true);

ALTER TABLE libro CHANGE AñoPubli año_publi INT;
-- esto es para modificar la estructura de la tabla libro, cambiando la columna AñoPubli por año_publi. Debido a un error.

```

IMPLEMENTAR SOLUCIÓN

Una vez tengamos la Base de Datos creada, debemos conectarla a una aplicación para su desarrollo. Como ya hemos dicho anteriormente, utilizaremos la implementación de alquilar un

libro nuevo de la biblioteca. Esta mostrará una lista con los nombres de libros que ya han sido alquilados. añadiendo a la lista.

Para hacer eso debemos implementar nuestra base de datos, con la Tabla Libro para esta versión inicial. Así que cojeremos Libro.

```
CREATE TABLE libro (
    ID_libro SERIAL PRIMARY KEY,
    Titulo VARCHAR(200) NOT NULL,
    Autor VARCHAR(100) NOT NULL,
    Editorial VARCHAR(100),
    Genero VARCHAR(200),
    AñoPubli INTEGER,
    Disponible BOOLEAN
);
```

Para el desarrollo de la aplicación utilizaremos IntelliJ. ¿Pero por qué?



Lo hemos elegido debido a su **potencia y versatilidad**, ya que ofrece una amplia gama de herramientas que permiten trabajar eficientemente en una variedad de tecnologías, que hacen que el desarrollo de software sea más eficiente.

Esta especialmente optimizado para el **desarrollo en Java**, el principal lenguaje que utilizaremos y que ofrece una integración perfecta, proporcionando herramientas avanzadas para su codificación, depuración, refactorización y análisis. No solo con JAVA, sino también integra otros lenguajes y tecnologías, como *Kotlin*, *Groovy*, *Scala*, *JavaScript*, *TypeScript*, y más. Tiene un potente editor de código, herramientas de refactorización, pruebas unitarias, integración con sistemas de control de versiones y mucho más.

Para aplicar nuestra base de datos al proyecto que hemos creado y que mas tarde aplicaremos debemos ir a un apartado, llamado Application.properties

Esta parte del proyecto sirve para configurar algunas propiedades de la aplicación, estando relacionadas con la configuración de la base de datos, del servidor web embebido, de la seguridad y otras.

Application.properties

```
server.port=8081
spring.datasource.url=jdbc:mysql://localhost:3306/biblioteca
spring.datasource.username=root
spring.datasource.password=clase
spring.datasource.driver=com.mysql.cj.jdbc.Driver
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.MySQL8Dialect
```

```
spring.jpa.show-sql=true
```

En cuanto a la que nos interesa ahora, la configuración de la base de datos, se debe especificar:

- Puerto en que se ejecutara el servidor. En este caso el 8081. → `server.port=8081`
- La URL de conexión JDBC a la base de datos MySQL, biblioteca, con servidor 3306 →
`spring.datasource.url=jdbc:mysql://localhost:3306/biblioteca`
- El nombre de usuario de la BBDD al que la aplicación se conectará, siendo `root`
→ `spring.datasource.username=root`
- La contraseña del usuario de la bbdd MySQL, siendo clase →
`spring.datasource.password=clase`
- El Driver o controlador JDBC, utilizado para la conexión con la bbdd, utilizando MySQL Connector/J
→ `spring.datasource.driver=com.mysql.cj.jdbc.Driver`
- El dialecto de Hibernate, que será utilizado para generar consultas SQL de la bbdd.
→ `spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect`
- Por último, la propiedad que indica a Spring boot que muestre las consultas
→ `spring.jpa.show-sql=true`

DESARROLLO DE LA SOLUCIÓN

Empezamos a crear nuestra aplicación de una biblioteca en IntelliJ.

Para desarrollar nuestra aplicación utilizaremos **Spring Boot**. Marco de desarrollo de aplicaciones JAVA para simplificar la creación de aplicaciones empresariales. Proporciona un conjunto de características y herramientas para desarrollar aplicaciones de forma rápida y con poco esfuerzo. Spring Boot facilita la configuración automática y la creación de aplicaciones autónomas que se pueden ejecutar fácilmente sin necesidad de una configuración compleja. Ofrece características como *incorporación de servidores web embebidos, gestión de dependencias, configuración automática, pruebas integradas* y más, lo que hace que el desarrollo de aplicaciones Java sea más eficiente, productivo y rápido.

Para empezar abrimos nuestra aplicación IntelliJ y creamos nuestro proyecto.

Debemos aplicar ciertas dependencias:

1) spring-boot-starter-data-jpa:

Data JPA ⇒ Esta dependencia integrada en la aplicación, simplifica el acceso a los datos utilizando JPA (*Java Persistence API*) y ofrece consultas declarativas, paginación, y soporte para transacciones.

2) spring-boot-starter-thymeleaf:

Thymeleaf ⇒ Motor de plantillas Java XML/HTML, para crear vistas web en aplicaciones Spring. Esto renderiza vistas HTML en la aplicación web.

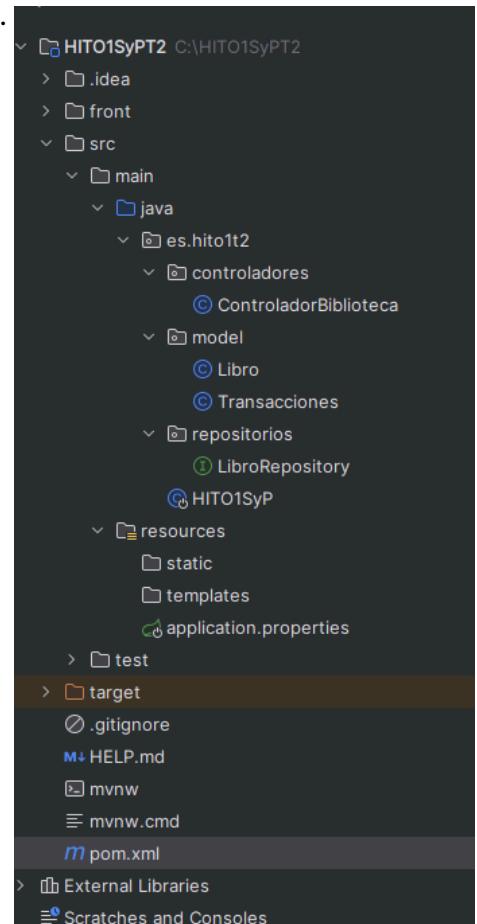
3) spring-boot-starter-web:

Web ⇒ Con esta dependencia se desarrollan aplicaciones web utilizando *Spring MVC* (Model-View-Controller). Configura servidores web embebidos y maneja las peticiones HTTP, así como la integración con otras tecnologías web de Spring.

4) spring-boot-starter-test:

Test ⇒ Esta dependencia incluye bibliotecas para escribir pruebas unitarias y de integración en la aplicación *Spring Boot*. Proporciona soporte para *JUnit*, *Mockito* y otras herramientas de pruebas en Java.

5) mysql-connector-java: Esta dependencia es un conector JDBC para MySQL, que permite a la aplicación comunicarse con una base de datos MySQL utilizando el protocolo JDBC (Java Database Connectivity).



En el Back-end nos partimos el trabajo en partes.

1. Controlador
2. Model
3. Repositorio
4. Application.properties, aunque ya la explicamos
5. Pom.xml

BACK-END

Vamos a empezar por lo último: El POM.

5. POM:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.0.1</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>es.hito1sp</groupId>
```

```

<artifactId>HITO1SyPT2</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>Biblioteca</name>
<description>Aplicación de alquiler de libros</description>
<properties>
    <java.version>17</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.31</version>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>

```

Esta parte del proyecto describe cómo se construye un proyecto en *Maven* y sus dependencias necesarias. Vamos por partes:

`</modelVersion>` → Define la versión del modelo POM. Siendo esta la 4.0.0.

`</parent>` → El proyecto hereda la configuración del proyecto padre `spring-boot-starter-parent` de Spring Boot. Permite definir algunas configuraciones para los proyectos de Spring Boot.

```
<groupId>
<artifactId>
<version>
```

Identifican al proyecto.

El **groupid** ⇒ indica el grupo al que pertenece el proyecto.

artifactid ⇒ Nombre del proyecto

Version ⇒ Versión del proyecto.

```
<name> || <description>
```

Proporcionan una descripción y un nombre al proyecto.

`<properties>` → Define propiedades del proyecto, como la versión de Java a 17.

```
<dependencies>
```

Dependencias: Aquí se enumeran las dependencias del proyecto. Cada dependencia tiene un **groupid** y un **artifactid** que identifican la biblioteca que se está utilizando.

- a. `spring-boot-starter-data-jpa`:
- b. `spring-boot-starter-thymeleaf`:
- c. `spring-boot-starter-web`:
- d. `spring-boot-starter-test`:
- e. `mysql-connector-java`:

Build (Construcción): Define cómo se debe construir el proyecto. En este caso, se utiliza el plugin `spring-boot-maven-plugin` para empaquetar la aplicación como un JAR ejecutable de Spring Boot.

```
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
```

1. CONTROLADOR: ControladorBiblioteca

```
package es.hito1t2.controladores;

import es.hito1t2.model.Libro;
```

```
import es.hito1t2.repositorios.LibroRepository;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;
@RestController
@CrossOrigin(origins = "*")

//config.addAllowedHeader("*"); // Permitir todos los encabezados
//    config.addAllowedMethod("*");
@RequestMapping("/biblioteca")
public class ControladorBiblioteca {

    @Autowired
    private LibroRepository libro;

    @GetMapping("/todos")
    public List<Libro> obtenerTodosLosLibros() {
        return libro.findAll();
    }

    @GetMapping("/{id}")
    public Libro obtenerLibroPorId(@PathVariable("id") int id) {
        return libro.findById(id).orElse(null);
    }

    @PostMapping("/nuevo")
    public Libro crearNuevoLibro(@RequestBody Libro nuevoLibro) {
        libro.save(nuevoLibro);
        return nuevoLibro;
    }

    @PutMapping("/modificar/{id}")
    public Libro modificarLibro(@RequestBody Libro libroModificado,
@PathVariable("id") int id) {
        libroModificado.setId(id);
        libro.save(libroModificado);
        return libroModificado;
    }

    @DeleteMapping("/eliminar/{id}")
    public String eliminarLibro(@PathVariable("id") int id) {
        libro.deleteById(id);
    }
}
```

```
        return "Libro con ID " + id + " eliminado";
    }
}
```

Esta parte es donde se encuentra el controlador de la aplicación web construida con Spring Boot. Se encarga de gestionar las operaciones relacionadas con la biblioteca de libros.

Importaciones:

- Importamos las clases necesarias para el funcionamiento del controlador, como clases de Spring Framework para la gestión de solicitudes HTTP.

Anotaciones:

- **@RestController** ⇒ Indica que esta clase es un controlador Spring MVC que maneja solicitudes web y devuelve datos en formato JSON.
- **@CrossOrigin(origins = "*")** ⇒ Permite solicitudes de cualquier origen (CORS) para acceder a los recursos de este controlador *desde cualquier dominio*.

Mapeo de URL:

- **@RequestMapping("/biblioteca")** ⇒ Mapea las solicitudes que comienzan con /biblioteca a este controlador.

Inyección de dependencias:

- **@Autowired**
private LibroRepository libro ⇒ Inyecta una instancia de LibroRepository en el controlador, permitiendo acceder a otros métodos para interactuar con la base de datos.

Métodos del controlador:

- **obtenerTodosLosLibros()** → Maneja las solicitudes GET en **/biblioteca/todos** para devolver una lista de todos los libros de la biblioteca.
- **obtenerLibroPorId(int id)** → Maneja solicitudes GET en **/biblioteca/{id}** y devuelve el libro correspondiente al ID proporcionado.
- **crearNuevoLibro(Libro nuevoLibro)** → Maneja solicitudes POST en **/biblioteca/nuevo** para crear un nuevo libro.
- **modificarLibro(Libro libroModificado, int id)** → Maneja las solicitudes PUT en **/biblioteca/modificar/{id}** para modificar un libro existente.
- **eliminarLibro(int id)** → Maneja solicitudes DELETE en **/biblioteca/eliminar/{id}** para eliminar un libro por su ID.

Estos son operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los libros en la biblioteca, interactuando con el repositorio de libros (LibroRepository) para realizar operaciones de persistencia en la base de datos.

DIFERENCIAS entre las solicitudes HTTP **GET / POST / PUT / DELETE** para el diseño de aplicaciones web y servicios RESTful.

GET == Sigue datos del servidor.

- Método seguro, que NO modifica los datos del servidor y puede repetirse sin efectos secundarios.
- Los parámetros de la solicitud se envían en la URL.
- Se usa para recuperar recursos o datos del servidor.
- **Ejemplo:** Obtener información de un usuario con una URL GET **/usuarios/1**.

POST == Enviar datos al servidor para crear o cambiar recursos.

- Puede modificar los datos del servidor. Realizar la misma solicitud varias veces puede tener efectos diferentes.
- Los datos se envían en el cuerpo de la solicitud HTTP.
- Se usa para enviar datos del cliente al servidor, como formularios, carga de archivos, etc
- **Ejemplo:** Crear un nuevo usuario enviando sus datos al servidor con una solicitud POST a [/usuarios](#).

PUT == Actualizar recursos existentes en el servidor.

- NO modifica los datos del servidor. Realizar la misma solicitud varias veces no produce efectos secundarios.
- Los datos se envían en el cuerpo de la solicitud HTTP.
- Actualizar un recurso o lo crea si no existe.
- **Ejemplo:** Actualizar la información de un usuario existente con una solicitud PUT a [/usuarios/1](#).

DELETE == Eliminar recursos existentes en el servidor.

- NO modifica los datos del servidor. Realizar la misma solicitud varias veces no produce efectos diferentes.
- No tiene cuerpo de solicitud, ya que la identificación del recurso se incluye en la URL.
- **Ejemplo:** Eliminar un usuario existente con una solicitud DELETE a [/usuarios/1](#).

2. Model: LIBRO

```
package es.hito1t2.model;

import jakarta.persistence.*;
import java.io.Serializable;

@Entity
public class Libro implements Serializable {
    @Column(name = "ID_libro")
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String titulo;
    private String autor;
    private String editorial;
    private String genero;
    @Column(name = "año_publi")
    private int añoPubli;

    private Boolean disponible;

    public Libro() { }
```

```
public Libro(int id, String titulo, String autor, String editorial, String genero, int añoPubli, Boolean disponible) {
    this.id = id;
    this.titulo = titulo;
    this.autor = autor;
    this.editorial = editorial;
    this.genero = genero;
    this.añoPubli = añoPubli;
    this.disponible = disponible;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getTitulo() {
    return titulo;
}

public void setTitulo(String titulo) {
    this.titulo = titulo;
}

public String getAutor() {
    return autor;
}

public void setAutor(String autor) {
    this.autor = autor;
}

public String getEditorial() {
    return editorial;
}

public void setEditorial(String editorial) {
    this.editorial = editorial;
}

public String getGenero() {
    return genero;
}

public void setGenero(String genero) {
```

```

        this.genero = genero;
    }

    public int getAñoPubli() {
        return añoPubli;
    }

    public void setAñoPubli(int añoPubli) {
        this.añoPubli = añoPubli;
    }

    public Boolean getDisponible() {
        return disponible;
    }

    public void setDisponible(Boolean disponible) {
        this.disponible = disponible;
    }
}

```

La clase Libro es una entidad de modelo en una aplicación Java que utiliza el framework de persistencia JPA.

La clase Libro define la estructura de un libro en el sistema, proporcionando métodos para acceder y modificar sus atributos. Cuando se utiliza en conjunción con JPA, esta clase se mapea a una tabla en la base de datos, lo que permite el almacenamiento y recuperación de instancias de libros.

Importaciones:

- `jakarta.persistence.*`: Importa las anotaciones y clases necesarias para definir entidades JPA.

Anotaciones:

- `@Entity` ⇒ Indica que esta clase es una entidad JPA y se debe mapear a una tabla en la base de datos.
- `@Column(name = "ID_libro")` ⇒ El campo id se mapea a la columna *ID_libro* en la base de datos.
- `@Id` ⇒ Indica que el campo id es la clave primaria deLibro.
- `@GeneratedValue(strategy = GenerationType.IDENTITY)` ⇒ El valor de la clave primaria se genera automáticamente por la base de datos.

Atributos:

- **id**: Identificador único del libro.
- **titulo**: Título del libro.
- **autor**: Autor del libro.
- **editorial**: Editorial del libro.
- **genero**: Género del libro.
- **añoPubli**: Año de publicación del libro.
- **disponible**: Indica si el libro está disponible o no.

Constructores:

- Tiene 2 constructores, uno vacío que pide la clase obligatoriamente, y otro que inicializa todos los atributos.

Métodos de acceso (Getters y Setters):

- Los métodos **get** y **set**, utilizados para acceder y modificar los atributos de la clase Libro, interactúan con los atributos de la clase desde otras partes.

3. Repositorio: LibroRepository

```
package es.hito1t2.repositorios;

import es.hito1t2.model.Libro;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface LibroRepository extends JpaRepository<Libro,
Integer> {

}
```

La clase Libro y la interfaz LibroRepository están relacionadas en cuanto a persistencia de datos en una aplicación Java, especialmente cuando se utiliza Spring Boot con Spring Data JPA. La clase Libro define la estructura y el comportamiento de un objeto libro, mientras que la interfaz LibroRepository define métodos para interactuar con la base de datos y realizar operaciones de persistencia sobre los libros. Esto está hecho de esta manera, para facilitar el manejo de la persistencia de datos en la aplicación.

LibroRepository es un repositorio de datos que define métodos para realizar operaciones de acceso a datos en la tabla de libros:

Extensión de JpaRepository:

- JpaRepository es una interfaz de Spring Data JPA con métodos predefinidos para realizar operaciones CRUD y otras consultas de la base de datos.
- **LibroRepository** extiende **JpaRepository<Libro, Integer>** ==> Opera en la entidad Libro, con Integer como tipo de dato para las claves primarias.

Anotaciones:

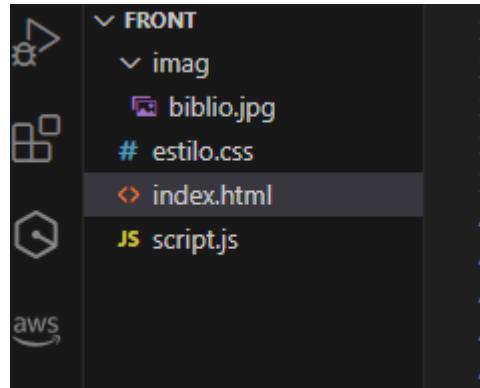
- **@Repository** ⇒ Lo marca como un componente de Spring, haciendo que Spring la escaneé y la administre como un componente de repositorio de datos.

FRONT-END

Ahora toca el *front-end*, que no será realizado en la propia aplicación de IntelliJ, si no que será creado, tanto el HTML, CSS como el JavaScript en *Visual Studio Code*. Esto es porque ya estoy acostumbrado a utilizarla y me es mas familiar.

Tendrá una estructura con HTML, CSS, para diseñar la web, y JavaScript, para definir sus funciones.

1. HTML
2. CSS
3. JavaScript



1. HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Biblioteca</title>
    <script src="script.js" defer></script>
    <link rel="stylesheet" type="text/css" href="estilo.css"> <!-- Enlazar
archivo CSS -->

</head>
<body>
    <div class="fondo">
        <h1 class="titulo">Biblioteca</h1>
    </div>
    <div class="separador"></div>
    <br>
    <h2 class="centro">Detalles del Libro</h2>
    <div id="detallesLibro"></div>

    <div class="centro">
        <label for="titulo"><b>Título:</b></label>
        <input type="text" id="titulo" name="titulo">
        <br>
        <br>
```

```

<label for="autor"><b>Autor:</b></label>
<input type="text" id="autor" name="autor">
<br>
<br>
<label for="editorial"><b>Editorial:</b></label>
<input type="text" id="editorial" name="editorial">
<br>
<br>
<label for="genero"><b>Género:</b></label>
<input type="text" id="genero" name="genero">
<br>
<br>
<label for="añoPubli"><b>Año de Publicación:</b></label>
<input type="text" id="añoPubli" name="año">
<br>
<br>
<label for="disponible"><b>Disponible:</b></label>
<input type="checkbox" id="disponible" name="disponible">
<br>
<br>
<button onclick="crearLibro()" class="botones">Añadir Libro</button>
<br>

</div>
<div class="separador"></div>
<br>
<h2 class="centro">Listado de Libros disponibles</h2>
<ul id="listaLibros"></ul>
<div class="centro">
<button onclick="mostrarLibros()" class="botones">Mostrar libros</button>
<table id="tablaLibros" class="tabla">
    <!-- Aquí se mostrarán los libros -->
</table>
</div>

</div>

</body>
</html>

```

2. CSS:

```
.fondo {
```

```
background-image: url('./imag/biblio.jpg'); /* Ruta a tu imagen de fondo */
*/
background-size: cover; /* Ajusta la imagen al tamaño del div */
background-position: center; /* Centra la imagen */
padding: 20px; /* Ajusta el espacio alrededor del contenido dentro del div */
*/
text-align: center; /* Centra el contenido horizontalmente */
height: 500px;
}

.titulo {
    color: white; /* Color del texto */
    font-size: 50px; /* Tamaño de la fuente */
}

.centro{
    text-align: center;
}

.botones{
    background-color: cornflowerblue; /* Color de fondo */
    border: none; /* Sin borde */
    color: white; /* Color del texto */
    padding: 10px 20px; /* Espaciado interno */
    text-align: center; /* Alineación del texto */
    text-decoration: none; /* Sin subrayado */
    display: inline-block;
    font-size: 16px; /* Tamaño de la fuente */
    font-family: Georgia, 'Times New Roman', Times, serif;
    margin: 4px 2px;
    cursor: pointer; /* Cursor al pasar sobre el botón */
    border-radius: 8px; /* Borde redondeado */
}

.separador {
    height: 10px; /* Altura de la línea */
    background-color: #000; /* Color de la línea */
    margin: 20px 0; /* Margen superior e inferior para separación */
}

.tabla{
    text-align: center;
```

```
    position: absolute;
    left: 20%;
}
```

Se crean eventos como **onclick** para llamar a funciones JavaScript cuando se haga clic en los botones. Estos métodos, como son **crearLibro()** y **mostrarLibros()** los veremos en la parte de JavaScript.

Enlace entre HTML y CSS:

```
<link rel="stylesheet" type="text/css" href="estilo.css">
```

Dentro del <head>.

Esto indica al navegador que cargue y aplique las reglas de estilo definidas en el archivo **estilo.css** a los elementos HTML de la página.

Enlace entre HTML y JavaScript:

```
<script src="script.js" defer></script>
```

Dentro del <head>.

Enlace entre HTML y nuestra Aplicación:

Este HTML se une a nuestra aplicación debido a que los archivos del proyecto se organizan en una estructura de directorios dentro de la aplicación.

3. JAVASCRIPT:

```
// Función para crear un nuevo libro
function crearLibro() {
    const titulo = document.getElementById('titulo').value;
    const autor = document.getElementById('autor').value;
    const editorial = document.getElementById('editorial').value;
    const genero = document.getElementById('genero').value;
    const año = document.getElementById('añoPubli').value;
    const disponible = document.getElementById('disponible').checked;

    const nuevoLibro = {
        titulo: titulo,
        autor: autor,
        editorial: editorial,
        genero: genero,
        añoPubli: año,
        disponible: disponible
    };
}
```

```
fetch(`http://localhost:8081/biblioteca/nuevo`, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(nuevoLibro)
})
.then(response => response.json())
.then(data => {
  console.log('Libro creado:', data);
  cargarLibros(); // Recargar lista de libros después de la creación

  // Limpiar los campos del formulario después de agregar el libro
  document.getElementById('titulo').value = '';
  document.getElementById('autor').value = '';
  document.getElementById('editorial').value = '';
  document.getElementById('genero').value = '';
  document.getElementById('añoPubli').value = '';
  document.getElementById('disponible').checked = false;

})
.catch(error => console.error('Error al crear libro:', error));
}

// Función para actualizar un libro existente
function cargarLibros() {
  const options = {
    method: "GET",
  };

  fetch(`http://localhost:8081/biblioteca/todos`, options)
    .then(response => response.json()) // Parsea la respuesta como JSON
    .then(data => llenarTabla(data))
    .catch(error => console.error('Error al cargar libros:', error)); // Maneja errores
}

function actualizarLibro(id) {
  let nuevaTitulo = prompt("Nuevo titulo:");
  let nuevoAutor = prompt("Nuevo autor:");
  let nuevoEditorial = prompt("Nueva editorial:");
  let nuevoGenero = prompt("Nuevo genero:");
}
```

```
let nuevoAño = prompt("Nuevo año de publicación:");

let libro = {
    "id": id,
    "titulo": nuevaTitulo,
    "autor": nuevoAutor,
    "editorial": nuevoEditorial,
    "genero": nuevoGenero,
    "añoPubli": parseInt(nuevoAño)
};

const options = {
    method: "PUT",
    headers: {
        'Content-Type': 'application/json'
    },
    body: JSON.stringify(libro)
};

fetch(`http://localhost:8081/biblioteca/modificar/${id}`, options)
    .then(response => {
        if (response.ok) {
            cargarLibros();
            mostrarLibros();
        } else {
            throw new Error('Error al actualizar libro');
        }
    })
    .catch(error => console.error('Error al actualizar libro:', error));
// Maneja errores
}

function eliminarLibro(id) {
    // Realizar una solicitud GET para obtener los detalles del libro
    fetch(`http://localhost:8081/biblioteca/${id}`)
        .then(response => response.json())
        .then(libro => {
            // Mostrar los detalles del libro al usuario
            const confirmacion = confirm(`¿Seguro que deseas eliminar el libro
"${libro.titulo}" de "${libro.autor}"?`);

            if (confirmacion) {

```

```
// Se crea la solicitud DELETE con el ID del libro a eliminar
const options = {
    method: "DELETE"
};

// Se realiza la solicitud DELETE al servidor
fetch(`http://localhost:8081/biblioteca/eliminar/${id}`, options)
    .then(response => {
        // Se verifica si la solicitud se completó correctamente
        if (response.ok) {
            mostrarLibros();
            // Si la eliminación fue exitosa, recargar la lista de
            libros
            cargarLibros();
        } else {
            // Si hubo un error en la eliminación, mostrar un
            mensaje de error
            throw new Error('Error al eliminar libro');
        }
    })
    .catch(error => console.error('Error al eliminar libro:', error));
// Manejar errores
}
.catch(error => console.error('Error al obtener detalles del libro:', error));
// Manejar errores
}

function mostrarLibros() {
    fetch('http://localhost:8081/biblioteca/todos')
        .then(response => response.json()) // Convertir la respuesta a JSON
        .then(libros => {
            // Obtener el elemento HTML donde se mostrarán los libros (por
            ejemplo, una tabla)
            const tablaLibros = document.getElementById('tablaLibros');

            // Limpiar la tabla de libros antes de agregar nuevos
            tablaLibros.innerHTML = '';

            // Iterar sobre la lista de libros y agregar cada uno a la tabla
            libros.forEach(libro => {
                // Crear una nueva fila para el libro
```

```

const fila = document.createElement('tr');

// Llenar la fila con los detalles del libro
fila.innerHTML =
    <td>${libro.titulo}</td>
    <td>${libro.autor}</td>
    <td>${libro.editorial}</td>
    <td>${libro.genero}</td>
    <td>${libro.añoPubli}</td>
    <td>${libro.disponible ? 'Sí' : 'No'}</td>
    <td>
        <button
            onclick="actualizarLibro(${libro.id})">Modificar</button>
        <button
            onclick="eliminarLibro(${libro.id})">Eliminar</button>
    </td>
`;

// Agregar la fila a la tabla
tablaLibros.appendChild(fila);
});

})
.catch(error => console.error('Error al mostrar libros:', error)); // Manejar errores
}

```

Estas funciones en JavaScript trabajan juntas para permitir la interacción del usuario con la aplicación web, realizando operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en los libros almacenados en la bbdd y actualizando la interfaz de usuario para mostrar los cambios:

1. crearLibro()

Crea un nuevo libro y lo agrega a la base de datos. ¿Como?:

- 1º → Obtiene los valores de los campos del formulario HTML (*título, autor, editorial, etc.*).
- 2º → Construye un objeto JavaScript que representa el nuevo libro utilizando los valores obtenidos (**nuevolibro**)
- 3º → Realiza una solicitud **POST** al servidor para agregar el nuevo libro a la base de datos con el método **fetch** para realizar solicitudes HTTP desde el cliente, es decir el navegador web, hacia un servidor web y manejar las respuestas obtenidas.
- 4º → Despues de agregar el libro con éxito:
 - Recarga la lista de libros llamando a *cargarLibros()*.
 - Limpia los campos del formulario.

2. cargarLibros()

Carga la lista de libros desde la base de datos/servidor y la muestra en la interfaz de usuario.

1º → Realiza una solicitud **GET** para obtener la lista de libros.

2º → Una vez que recibe la lista, llama a la función **llenarTabla(data)** para mostrar los libros en la interfaz.

3º → Maneja cualquier error que ocurra durante la solicitud, con:

```
.catch(error => console.error('Error al actualizar libro:', error));
```

3. actualizarLibro(id)

Actualiza los detalles de un libro, que ya este en la lista.

- **Parámetros:** → **id** del libro que se va a actualizar.

1º → Solicita al usuario los nuevos detalles del libro a través de prompts (nuevos títulos, autores, etc.).

2º → Construye un objeto JavaScript con los nuevos detalles del libro.

3º → Realiza una solicitud **PUT** para actualizar el libro con los nuevos detalles.

4º → Despues de actualizar los detalles del libro determinado:

- Recarga la lista de libros llamando a **cargarLibros()**.
- Muestra los libros actualizados en la interfaz.

4. eliminarLibro(id)

Elimina un libro existente del sistema.

- **Parámetros** → **id** del libro que se va a eliminar.

1º → Obtiene los detalles del libro mediante una solicitud **GET** a la bbdd.

2º → Muestra un mensaje de confirmación para confirmar la eliminación del libro.

3º → Si se confirma la eliminación → Realiza una solicitud **DELETE** para eliminar el libro.

4º → Despues de eliminar el libro:

- Recarga la lista de libros llamando a **cargarLibros()**.
- Muestra los libros actualizados en la interfaz.

5. mostrarLibros()

Muestra la lista de todos los libros en la interfaz de usuario.

1º → Realiza una solicitud **GET** al servidor para obtener la lista de libros.

2º → Limpia la tabla de libros en la interfaz.

3º → Para cada libro obtenido:

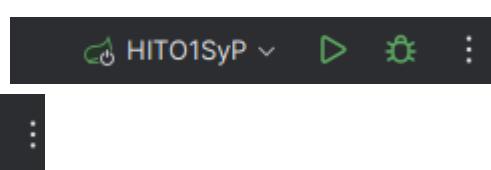
- Crea una fila en la tabla de libros y muestra los detalles del libro.
- Agrega botones para modificar y eliminar el libro.

4º → Maneja cualquier error que ocurra durante la solicitud.

EVALUACIÓN

Vamos a Evaluar nuestra aplicación.

Vamos a ejecurla desde IntelliJ: Le damos al botón



Y ya empieza.

La aplicación nos da cierta ruta, para enseñarnos la interfaz <http://127.0.0.1:5500/index.htm>
Donde se nos abrirá esta web:

The screenshot shows a large, ornate library hall with tall arched ceilings and rows of bookshelves. Overlaid on the image is a "Biblioteca" logo. Below the image is a "Detalles del Libro" (Book Details) form. The form fields include: Título: [text input], Autor: [text input], Editorial: [text input], Género: [text input], Año de Publicación: [text input], Disponible: [checkbox], and a blue "Añadir Libro" (Add Book) button. Below the form is a "Listado de Libros disponibles" (List of Available Books) section with a "Mostrar libros" (Show books) button.

Comprobamos que los métodos funcionan:

1. Añadir libro / crear Libro: [Añade el libro y limpia las entradas de texto](#)



Detalles del Libro

Título:

Autor:

Editorial:

Género:

Año de Publicación:

Disponible:

[Añadir Libro](#)

2. Mostrar Lista de libros.

Ahora, vamos al botón "Mostrar Libros", para saber si se nos muestra la lista y para saber si se han añadido a ella.

Listado de Libros disponibles

Mostrar libros					
El nombre del viento	Patrick Rothfuss	DAW Books	Fantasia	2007 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El temor de un hombre sabio	Patrick Rothfuss	DAW Books	Fantasia	2011 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El Imperio Final	Brandon Sanderson	Tor Books	Fantasia	2006 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El pozo de la ascensión	Brandon Sanderson	Tor Books	Fantasia	2007 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El héroe de las eras	Brandon Sanderson	Tor Books	Fantasia	2008 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Rebelión en la Bounty	John Boyne	C.S. Forester	Histórica	1932 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Cien años de soledad	Gabriel García Márquez	Editorial Sudamericana	Realismo mágico	1967 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El señor de los anillos: La comunidad del anillo 1984	J.R.R. Tolkien	Allen & Unwin	Fantasia	1954 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Matar a un ruiseñor	George Orwell	Secker & Warburg	Ciencia ficción	1949 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Harry Potter y la piedra filosofal	JK. Rowling	Bloomsbury Publishing	Novela	1960 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Orgullo y prejuicio	Jane Austen	T. Egerton, Whitehall	Fantasia	1997 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
La sombra del viento	Carlos Ruiz Zafón	Planeta	Novela romántica	1813 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Crónica de una muerte anunciada	Gabriel García Márquez	Editorial Sudamericana	Misterio	2001 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El alquimista	Paulo Coelho	HarperCollins	Novela	1981 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El principito	Antoine de Saint-Exupéry	Reynal & Hitchcock	Ficción	1988 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El código Da Vinci	Dan Brown	Doubleday	Literatura infantil	1943 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Crimen y castigo	Fyodor Dostoevsky	The Russian Messenger	Ficción de misterio	2003 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
La ciudad y los perros	Mario Vargas Llosa	Seix Barral	Novela psicológica	1866 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
Don Quijote de la Mancha	Miguel de Cervantes	Juan de la Cuesta	Novela satírica	1963 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
	Ken Follett	William Morrow and company	Historico	1605 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
				1989 Si	<input type="button"/> Modificar <input type="button"/> Eliminar

3. Modificar lista:

En la propia lista, encontramos dos botones. Uno de ellos es el botón "*Modificar*", que al pulsarlo, se nos abre un apestañita arriva para añadir los nuevos datos.

127.0.0.1:5500 dice

Nuevo título:

|

Aceptar Cancelar

Disponible:

Añadir Libro

Listado de Libros disponibles

Mostrar libros					
El nombre del viento	Patrick Rothfuss	DAW Books	Fantasia	2007 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El temor de un hombre sabio	Patrick Rothfuss	DAW Books	Fantasia	2011 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El Imperio Final	Brandon Sanderson	Tor Books	Fantasia	2006 Si	<input type="button"/> Modificar <input type="button"/> Eliminar
El pozo de la ascensión	Brandon Sanderson	Tor Books	Fantasia	2007 Si	<input type="button"/> Modificar <input type="button"/> Eliminar

En el que añadiremos el nuevo título que queremos ponerle al libro.

127.0.0.1:5500 dice

Nuevo título:

ttttttt

Aceptar

Cancelar

Al cambiar todos los detalles, o los que deseas, pulsa "Aceptar" y el libro se modificará, y la lista se actualizará. [Modificar libro y actualizar lista](#)

Como vemos aquí ya está actualizado:

El nombre del viento	Patrick Rothfuss	DAW Books	Fantasía	2007	Si	Modificar	Eliminar
El temor de un hombre sabio	Patrick Rothfuss	DAW Books	Fantasía	2011	Si	Modificar	Eliminar
El Imperio Final	Brandon Sanderson	Tor Books	Fantasía	2006	Si	Modificar	Eliminar
El pozo de la ascensión	Brandon Sanderson	Tor Books	Fantasía	2007	Si	Modificar	Eliminar
El héroe de las eras	Brandon Sanderson	Tor Books	Fantasía	2008	Si	Modificar	Eliminar
Rebelión en la Bounty	John Boyne	C.S. Forester	Histórica	1932	Si	Modificar	Eliminar
Cien años de soledad	Gabriel García Márquez	Editorial Sudamericana	Realismo mágico	1967	Si	Modificar	Eliminar
r de los anillos: La comunidad del anillo 1984	J.R.R. Tolkien George Orwell	Allen & Unwin Secker & Warburg	Fantasía Ciencia ficción	1954 1949	Si	Modificar	Eliminar
Matar a un ruiseñor	Harper Lee	J.B. Lippincott & Co.	Novela	1960	Si	Modificar	Eliminar
Harry Potter y la piedra filosofal	J.K. Rowling	Bloomsbury Publishing	Fantasía	1997	Si	Modificar	Eliminar
Orgullo y prejuicio	Jane Austen	T. Egerton, Whitehall	Novela romántica	1813	Si	Modificar	Eliminar
La sombra del viento	Carlos Ruiz Zafón	Planeta	Misterio	2001	Si	Modificar	Eliminar
Crónica de una muerte anunciada	Gabriel García Márquez	Editorial Sudamericana	Novela	1981	Si	Modificar	Eliminar
El alquimista	Paulo Coelho	HarperCollins	Ficción	1988	Si	Modificar	Eliminar
El principito	Antoine de Saint-Exupéry	Reynal & Hitchcock	Literatura infantil	1943	Si	Modificar	Eliminar
El código Da Vinci	Dan Brown	Doubleday	Ficción de misterio	2003	Si	Modificar	Eliminar
Crimen y castigo	Fyodor Dostoevsky	The Russian Messenger	Novela psicológica	1866	Si	Modificar	Eliminar
La ciudad y los perros	Mario Vargas Llosa	Seix Barral	Novela	1963	Si	Modificar	Eliminar
Don Quijote de la Mancha	Miguel de Cervantes	Juan de la Cuesta	Novela satírica	1605	Si	Modificar	Eliminar
ttttt	ttttt	ttttt	ttttt	3	No	Modificar	Eliminar

4. Eliminar libro de la lista.

Al pulsar el otro botón que se encuentra en la lista, más específicamente, el botón "eliminar", nos abrirá una página de advertencia. Nos preguntará si queremos eliminar el libro. Y al darle a "Aceptar" se eliminará y se actualizará la lista: [Eliminar libro y actualizar lista](#)

127.0.0.1:5500 dice

¿Seguro que deseas eliminar el libro "ttttt" de "ttttt"?

Aceptar

Cancelar

Año de Publicación:

Nombre del libro	Autor	Editorial	Categoría	Año	Sí	Modificar	Eliminar
El nombre del viento	Patrick Rothfuss	DAW Books	Fantasía	2007	Sí	Modificar	Eliminar
El temor de un hombre sabio	Patrick Rothfuss	DAW Books	Fantasía	2011	Sí	Modificar	Eliminar
El Imperio Final	Brandon Sanderson	Tor Books	Fantasia	2006	Sí	Modificar	Eliminar
El pozo de la ascensión	Brandon Sanderson	Tor Books	Fantasia	2007	Sí	Modificar	Eliminar
El héroe de las eras	Brandon Sanderson	Tor Books	Fantasia	2008	Sí	Modificar	Eliminar
Rebelión en la Bounty	John Boyne	C.S. Forester	Histórica	1932	Sí	Modificar	Eliminar
Cien años de soledad	Gabriel García Márquez	Editorial Sudamericana	Realismo mágico	1967	Sí	Modificar	Eliminar
señor de los anillos: La comunidad del anillo 1984	J.R.R. Tolkien	Allen & Unwin	Fantasia	1954	Sí	Modificar	Eliminar
Matar a un ruisenor	George Orwell	Secker & Warburg	Ciencia ficción	1949	Sí	Modificar	Eliminar
Harry Potter y la piedra filosofal	J.K. Rowling	Bloomsbury Publishing	Fantasía	1997	Sí	Modificar	Eliminar
Orgullo y prejuicio	Jane Austen	T. Egerton, Whitehall	Novela romántica	1813	Sí	Modificar	Eliminar
La sombra del viento	Carlos Ruiz Zafón	Planeta	Misterio	2001	Sí	Modificar	Eliminar
Crónica de una muerte anunciada	Gabriel García Márquez	Editorial Sudamericana	Novela	1981	Sí	Modificar	Eliminar
El alquimista	Paulo Coelho	HarperCollins	Ficción	1988	Sí	Modificar	Eliminar
El principito	Antoine de Saint-Exupéry	Reynal & Hitchcock	Literatura infantil	1943	Sí	Modificar	Eliminar
El código Da Vinci	Dan Brown	Doubleday	Ficción de misterio	2003	Sí	Modificar	Eliminar
Crimen y castigo	Fyodor Dostoevsky	The Russian Messenger	Novela psicológica	1866	Sí	Modificar	Eliminar
La ciudad y los perros	Mario Vargas Llosa	Seix Barral	Novela	1963	Sí	Modificar	Eliminar
Don Quijote de la Mancha	Miguel de Cervantes	Juan de la Cuesta	Novela satírica	1605	Sí	Modificar	Eliminar

CONCLUSIÓN

La gestión eficaz de los procesos empresariales y el uso ético de los datos son fundamentales para el éxito y la sostenibilidad de las organizaciones en un mundo cada vez más digitalizado.

A través de este trabajo, hemos explorado diversos aspectos relacionados con la optimización de los procesos, la consideración de implicaciones éticas y legales, y la aplicación de técnicas de ciencia de datos para mejorar la toma de decisiones empresariales.

La gestión de procesos empresariales no solo implica la optimización de flujos de trabajo, sino también la consideración de aspectos éticos y legales en el manejo de datos. El uso responsable de la información es crucial para preservar la confianza del cliente y garantizar el cumplimiento de regulaciones y normativas.

Además, la aplicación de técnicas de ciencia de datos ofrece oportunidades significativas para generar ideas valiosas a partir de conjuntos de datos complejos, apoyando así la toma de decisiones informadas y estratégicas en los negocios. Sin embargo, es fundamental que estas técnicas se utilicen de manera ética y responsable, evitando sesgos y asegurando la privacidad y la seguridad de los datos.

El éxito en la gestión de procesos empresariales y el aprovechamiento del potencial de los datos requieren un enfoque integrado que combine la eficiencia operativa con consideraciones éticas y un uso cuidadoso de las herramientas de ciencia de datos. Estos principios fundamentales son esenciales para construir organizaciones resilientes y adaptativas en un mundo empresarial en constante evolución.

BIBLIOGRAFÍA

[Curso de Java Fullstack Completo \(springboot, hibernate, JWT, API Rest\) \(youtube.com\)](#)
[Lucas Moy, 21 marzo 2020]

[El MEJOR IDE para Programar 🤔 ¿Visual Studio Code, IntelliJ...? Nuestro Editor de Código PREFERIDO \(youtube.com\)](#)
[Carlos Martín, 18 agosto 2015]