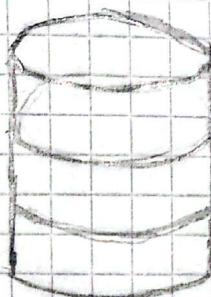
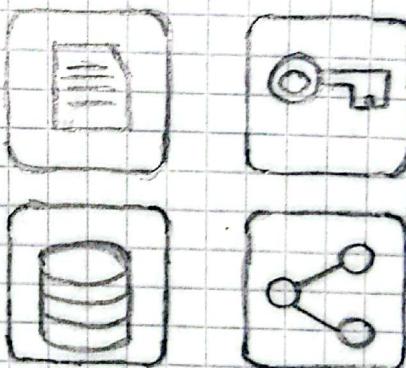


1 Estudio comparativo entre bases de datos relacionales y no relacionales.

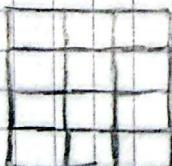
Este estudio comparativo analiza las bases de datos relacionales (SQL) y no relacionales (NoSQL), destacando que las primeras basadas en el modelo ACID, dominan el mercado actual debido a su robustez para el almacenamiento estructurado y transaccional, aunque presentan limitaciones en escalabilidad y flexibilidad. Por otro lado, las bases NoSQL, que operan bajo el modelo BASE, ofrecen mayor adaptabilidad para manejar datos no estructurados y grandes volúmenes de información en arquitecturas distribuidas, lo que las hace ideales para aplicaciones modernas y escalables. La elección entre ambas depende de factores como el tipo de datos, los requisitos de consistencia y la necesidad de escalabilidad horizontal o vertical. El documento concluye que si bien las bases relationales mantienen su relevancia en entornos estructurados, las no relationales están ganando terreno en escenarios que demandan agilidad y crecimiento flexible.



Relacionales (SQL)



No Relacionales
(NoSQL)



Bases de datos

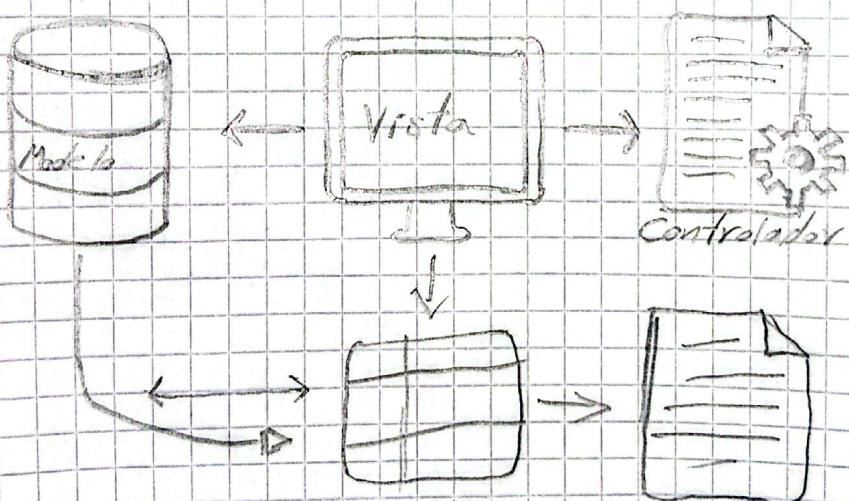
Reflexión

Según el artículo las bases relationales y no relationales, pienso que las bases relationales son mas estables para mi proyecto o mas adecuadas aunque las bases de datos no relationales son muy adecuadas para el manejo, mas no de informacion no lo veo prudente para el proyecto actual.

2 POO y MVC en PHP

Es una guía práctica dirigida a programadores que desean migrar de la programación estructurada al paradigma de programación orientada a objetos (POO) y aplicar el patrón arquitectónico MVC (Modelo-Vista-Controlador) en PHP. El libro está dividido en cuatro capítulos que tienen teoría, ejemplos prácticos, ejercicios de autoevaluación y proyectos reales. Explica conceptos fundamentales de POO como clases, objetos, métodos, propiedades, herencia, encapsulamiento y polimorfismo, así como la implementación de interfaces y clases abstractas. Luego introduce la arquitectura MVC, detallando el rol de modelo (lógica del negocio), la vista (interfaz de usuario) y el controlador (intermediario). Con ejemplos de código y una estructura de directorios recomendada. Incluye también sugerencias de estilo, metáforas para facilitar la comprensión y respuestas a preguntas frecuentes, terminando con un ejemplo completo de un sistema de ABO (Altas, Bajas y Modificaciones) de usuarios implementando POO y MVC en PHP.

(MVC)



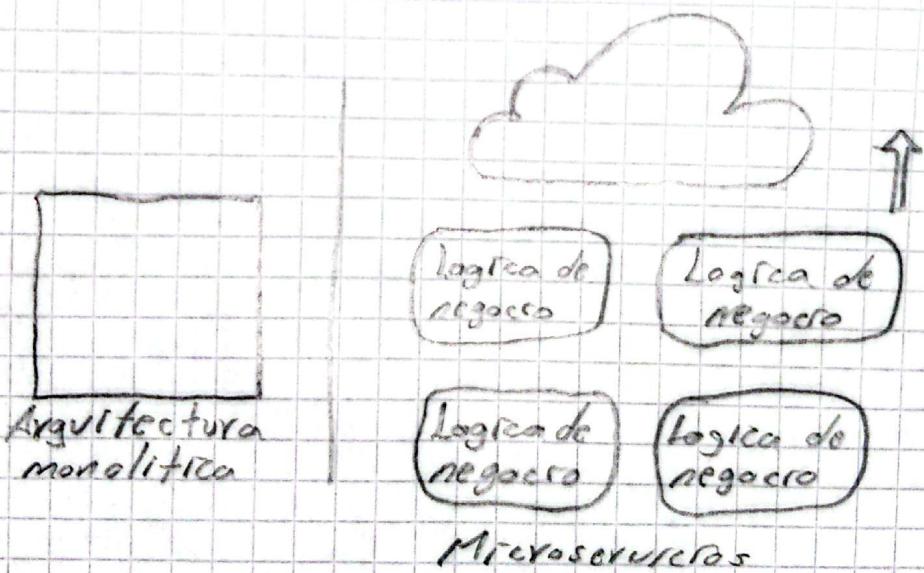
Reflexión

Aunque trabajo con Java y el documento está inspirado en PHP no importa mucho el criterio recíproco pero el lenguaje es. Cada controlador tiene su responsabilidad cada servicio está documentado. No se mezclan responsabilidades ni improviso estructuras. Con el patrón MVC bien aplicado en una improvisación de orden y eso no depende del lenguaje pero el punto es que este documento me ayudó a revisar el orden de mi proyecto.

3

Arquitectura de Microservicios

La arquitectura de microservicios es un estilo arquitectónico que organiza una aplicación como un conjunto de servicios pequeños, independientes y altamente cohesionados, cada uno ejecutándose en su propio proceso y comunicándose mediante APIs ligeras. Surge como respuesta a las limitaciones de las arquitecturas monolíticas, que, aunque simples inicialmente, se vuelven complejas y difíciles de mantener y escalar a medida que crecen. Los microservicios permiten escalar de forma granular, actualizar tecnologías específicas con mayor agilidad y organizar equipos alrededor de capacidades de negocio (Ley de Conway), favoreciendo la autonomía y la entrega continua. Cada servicio gestiona su propia base de datos, lo que posibilita el uso de diferentes tecnologías (SQL, NoSQL) según sus necesidades. El documento incluye un caso de estudio aplicado a tiendas de barrio en Colombia, donde se propone un microservicio para gestión de pedidos, ilustrando cómo este enfoque facilita la evolución iterativa y la adaptación tecnológica.

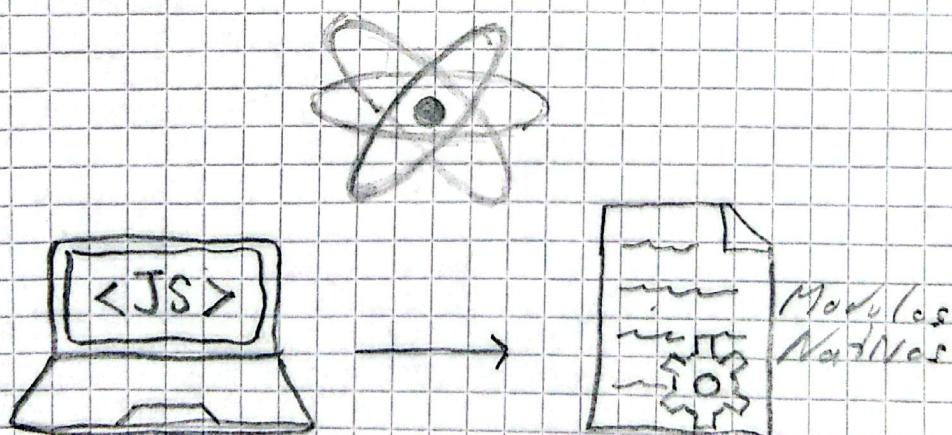


Reflexión

Este documento refirma algo que ya aplico en mi proyecto, bien así sea los microservicios muy buenas para mantener orden, trazabilidad y escalabilidad. según las microservicios permiten que cada parte del sistema evolucionen sin afectar el resto, y eso realmente me gustaría tener en mi proyecto pero el tiempo es escaso.

4 React Native: Aciertando las distancias entre desarrollo y diseño móvil multiplataforma

React native es un framework de código abierto que permite desarrollar aplicaciones móviles nativas para iOS y android utilizando un lenguaje Javascript. Surge como respuesta a la alta demanda de aplicaciones móviles y la necesidad de unificar el desarrollo entre plataformas, evitando la duplicación de esfuerzos al escribir código específico para cada sistema operativo. A diferencia de las aplicaciones híbridas (basadas en Web View) React native compila Javascript a código nativo, lo que resulta en un rendimiento y una experiencia de usuario similares a las aplicaciones desarrolladas nativamente con Java o Objective-C. Entre sus ventajas destacan la reutilización de hasta el 90% del código entre plataformas, una curva de aprendizaje accesible para desarrolladores web, herramientas de depuración eficientes y la capacidad de acceso a APIs nativas. Sin embargo, también presenta limitaciones, como la incapacidad para renderizar objetos 3D complejos o desarrollos videojuegos avanzados, y la necesidad de usar macOS para desarrollar en iOS. Se incluye un ejemplo práctico con la aplicación "Market Eat" demostrando su utilidad en el desarrollo ágil de interfaces nativas y su integración con tecnologías como Firebase. Se sabe que React Native es una opción robusta y cercana para el desarrollo móvil multiplataforma, especialmente para equipos con experiencias en Javascript y React.



Reflexión

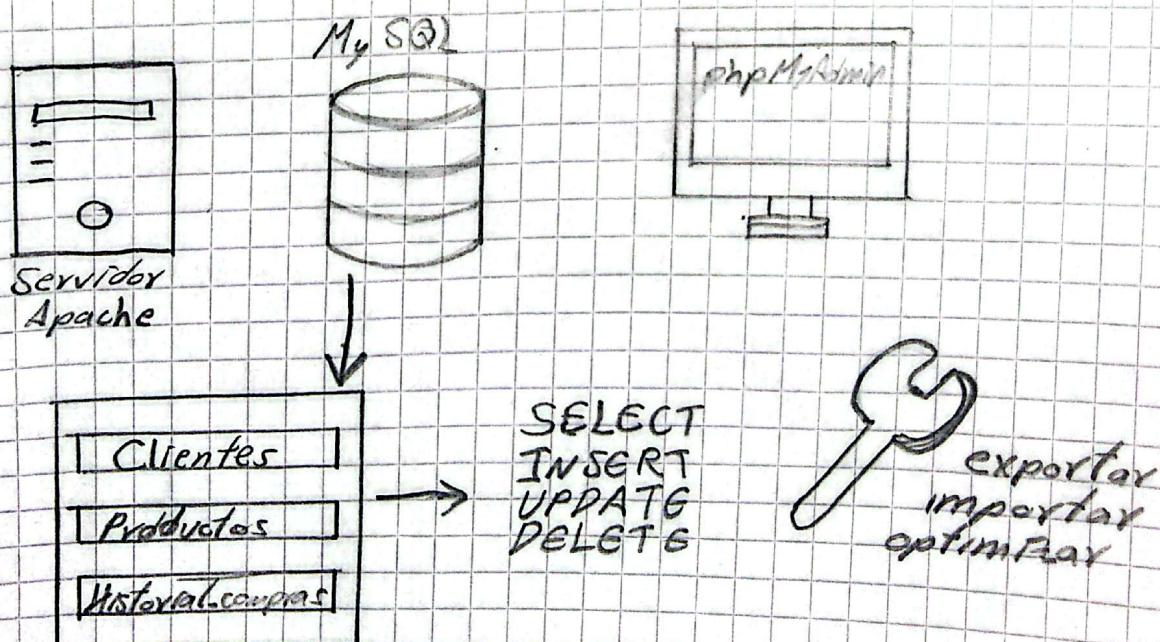
Pienso que React Native es una herramienta muy poderosa para desarrollar aplicaciones móviles multiplataforma con una sola base de código. En mi proyecto si se utiliza react

native para lograr hacer que nuestro proyecto tenga el
aplicativo móvil.

5

Creación y gestión de una base de datos con MySQL y PHPMyAdmin

Este es un trabajo de fin de grado que presenta una guía práctica para la creación y gestión de una base de datos utilizando MySQL y su interfaz gráfica de administración, phpMyAdmin. El trabajo surge de la necesidad de proporcionar una base de conocimiento accesible para usuarios sin experiencia previa en este ámbito, permitiéndoles gestionar los datos de un sitio web o proyecto de manera local. A diferencia de la mera teoría, este documento se centra en la aplicación práctica, guiando al lector desde la instalación del entorno (mediante el paquete WAMP, que incluye Apache, MySQL y PHP) hasta la ejecución de consultas SQL y las tareas de mantenimiento. Entre sus ventajas destacan el uso de tecnologías de código abierto y gratuitas, la sencillez de la interfaz de phpMyAdmin para realizar operaciones sin escribir código y la eficiencia de MySQL para manejar datos con pocos recursos. Sin embargo, también se advierte lo que para funciones avanzadas o entornos de gran escala se requeriría un conocimiento más profundo de SQL. El trabajo incluye un caso práctico completo con la creación de las tablas para una tienda online (clientes, productos, historial-compras), demostrando la utilidad de las bases de datos no relacionales. Se concluye que MySQL y PHPMyAdmin son herramientas potentes y accesibles que pueden impulsar cualquier negocio o idea en la era digital.

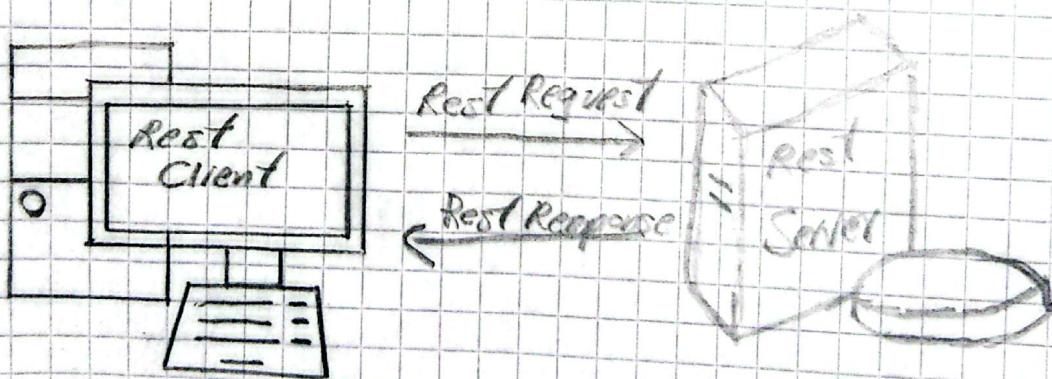


Reflexión:

Siendo pensando que las bases de datos SQL son modelos para mi proyecto en estas fases, este documento muestra como definir relaciones claras entre entidades aplicar restricciones de integridad y mantener consistencia en los datos., el documento me reafirma que el modelo relacional no es solo una elección técnica.

6 Historia familiar y Plataforma Web de Salud Integral. Práctica Social

Este proyecto presenta un sistema web desarrollado para la Facultad de enfermería de la Universidad de Antioquia, cuyo objetivo es gestionar y analizar información de salud familiar en contextos comunicados y rurales. La plataforma surge como respuesta a la falta de herramientas digitales eficaces para la sistematización de datos de Salud en zonas con acceso limitado a servicios, mejorando la capacidad de toma de decisiones basadas en evidencia. Desarrollada con Next.js en el frontend y Spring Boot en el Backend, utiliza una base de datos MySQL y sigue los principios de la arquitectura limpia para garantizar escalabilidad, mantenibilidad y un diseño desacoplado. La plataforma incluye módulos de seguridad con JWT, visualización de datos mediante gráficas interactivas y gestión de roles diferenciados para administradores (profesores) e interlocutores (estudiantes). Entre sus resultados destacan una interfaz intuitiva, dashboards analíticos y la capacidad de trabajar tanto en línea como fuera la línea gracias a su integración con una aplicación móvil en Flutter. El proyecto no solo moderniza la recolección y el análisis de datos en salud, sino que también fortalece la colaboración interdisciplinaria entre ingeniería y salud, cerrando brechas de inequidad en el acceso a la información sanitaria.

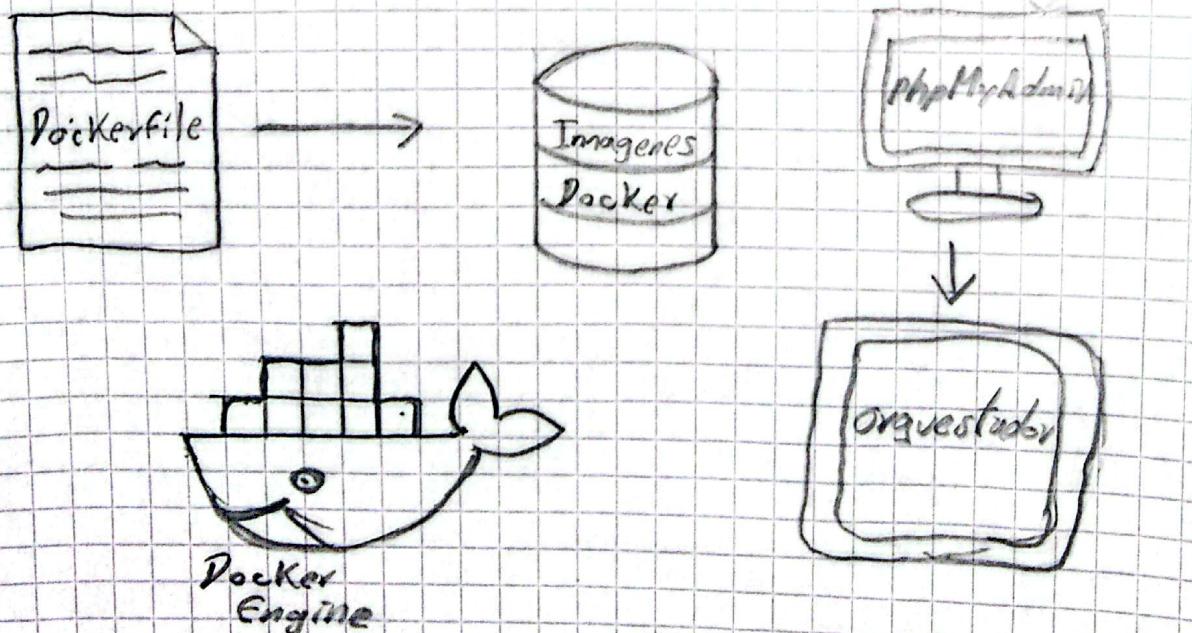


Reflexión:

En mi proyecto, aunque el foco está en la arquitectura Backend, esta reflexión me obliga a pensar en cómo se estructuran los datos sensibles, este documento me hizo pensar en la interoperabilidad, es decir la capacidad de plataformas o aplicaciones para comunicarse entre sí es decir compartir datos y entenderse mutuamente.

F Utilización de Docker para desplegar en producción sistemas de información heredados

El artículo presenta Docker como una solución eficaz para desplegar en producción sistemas de información heredados que enfrentan problemas de compatibilidad debido a versiones obsoletas de software, librerías o sistemas operativos. El caso de estudio describe la migración de un sistema desarrollado en PHP 7.4 desde una computadora personal a un entorno de producción en Rocky Linux 9, donde la versión requerida de PHP ya no es soportada. La virtualización tradicional resultó insuficiente debido a incompatibilidades en cascada, por lo que se optó por los contenedores con Docker. Esto permitió encapsular el sistema operativo y facilitando la creación de entornos aislados para desarrollo, pruebas y producción. Entre las ventajas clave se destacan la homologación de ambientes, la reducción de tiempos de despliegue (de días a horas), la optimización de recursos y la portabilidad de multiplataforma. Sin embargo, se reconoce una curva de aprendizaje asociada al manejo de contenedores y la necesidad de conocimientos en redes y seguridad. El artículo concluye que Docker es una tecnología escalable y cada vez más adoptada ideal para modernizar aplicaciones legacy sin comprometer su funcionalidad original.



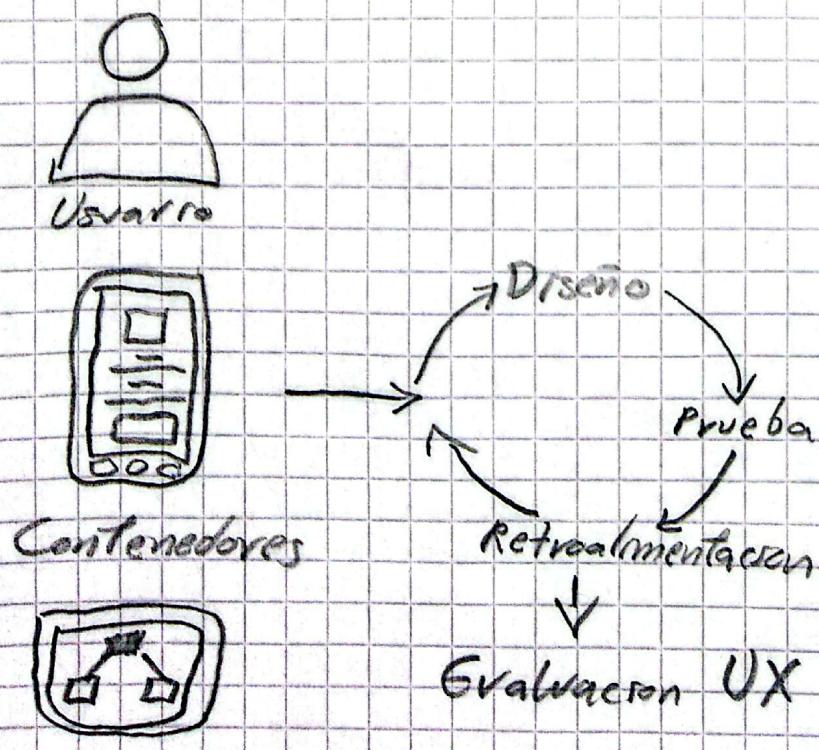
Reflexión:

Docker permite encapsular aplicaciones en contenedores que incluyen todo lo necesario para ejecutarlas en su Dockerfile.

permite definir exactamente como debe ejecutarse cada componente
sin sorpresas ni conflictos.

Evaluación de la experiencia del usuario (UX) en procesos gamificados: Una revisión bibliográfica

Este artículo aborda la evaluación de la experiencia de usuario en procesos gamificados, destacando la creciente relevancia de comprender como los usuarios interactúan con tecnologías que incorporan elementos de juego en contextos no lúdicos. El estudio identifica y analiza métodos, técnicas y herramientas utilizadas para medir la UX en entornos gamificados, con el objetivo de mejorar la inclusividad, accesibilidad y efectividad de estos sistemas. Mediante una revisión bibliográfica estructurada, se examinan indicadores que van desde la usabilidad y estética hasta la satisfacción y adaptabilidad cultural, recopilando enfoques evaluativos de estudios previos. Los resultados revelan una amplia variedad de métricas y prácticas como pruebas A/B, cuestionarios estandarizados y lineamientos como las heurísticas de Nielsen y las WCAG de la W3C. Se concluye que la evaluación de la UX en gamificación requiere enfoque multidisciplinarios y herramientas validadas para adaptarse a contextos específicos y se propone una guía práctica para integrar la evaluación UX en el diseño e implementación de procesos gamificados, dando importancia de la iteración y la mejora continua basada en datos reales de usuarios.

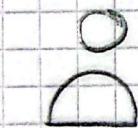


Reflexión:

El diseño UI suele asocarse directamente con el frontend pero el desarrollo me recuerda que la experiencia del usuario también depende profundamente del backend. Cada decisión técnica que tomo afecta como se siente el sistema desde el lado del usuario. UI no es solo estética o navegación.

9 Metodología SCRUM y desarrollo de repositorio digital

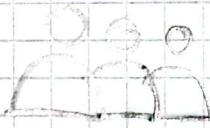
Se presenta el desarrollo de un repositorio digital para gestionar y difundir la productividad académica y científica de docentes e investigadores de la facultad de Contaduría y Administración de la Universidad Autónoma de Baja California. El repositorio se considera como una herramienta de acceso abierto que controliza y preserva recursos como artículos indexados, tesis, ponencias, patentes y materiales didácticos, siguiendo lineamiento de CONACYT y estándares informacionales de metadatos (Dublin Core). Para su desarrollo, se adoptó la metodología agil SCRUM, que permite una gestión flexible y colaborativa del proyecto, con entregas iterativas, priorización de funcionalidades y adaptación constante a los requerimientos del cliente. El equipo, compuesto por estudiantes de informática y docentes certificados en SCRUM, utilizaron herramientas como basecamp para facilitar la comunicación y el seguimiento. El repositorio se implementó con un sistema de gestión de contenidos (CMS) estructurado en un backend con base de datos relacional y un frontend accesible. El proyecto demuestra como la combinación de tecnologías de información y metodologías agiles optimiza la difusión del conocimiento académico, fomenta la colaboración y la visibilidad institucional.



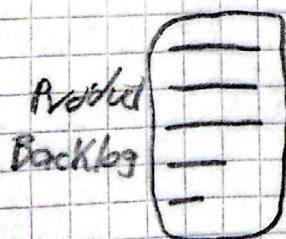
Product
Owner



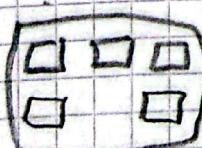
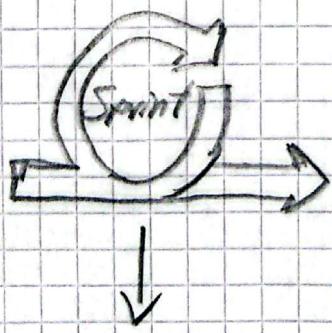
Scrum
Master



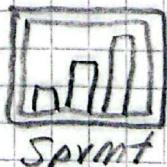
Equipo de
desarrollo



Product
Backlog



Sprint
Backlog



Sprint

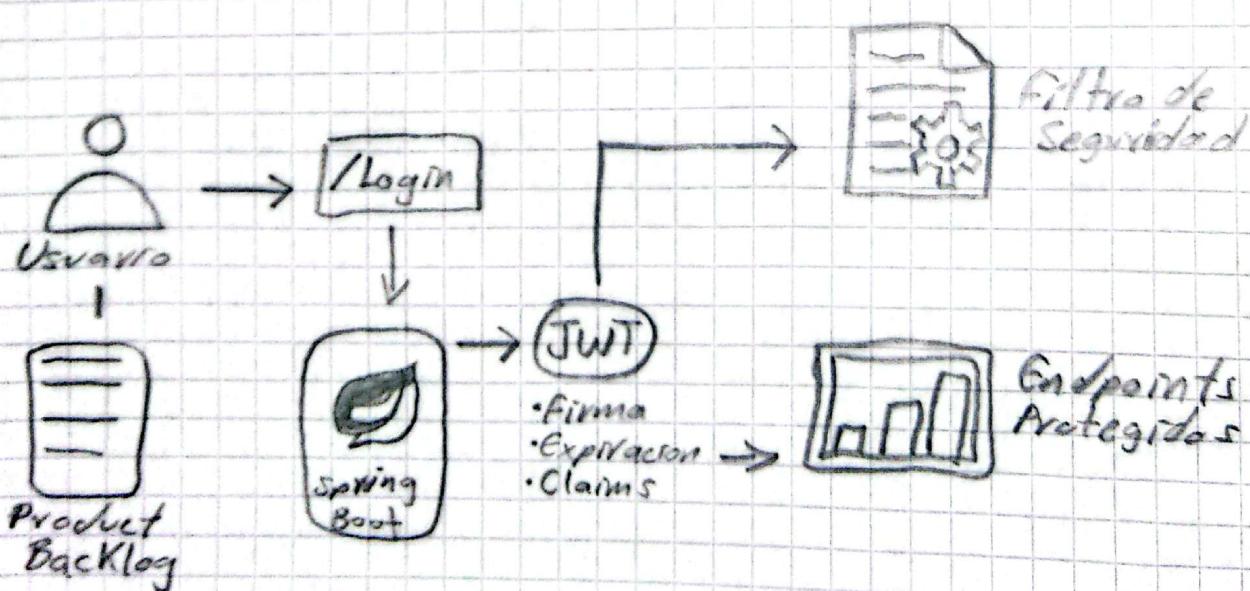
Reflexión:

Scrum tiene muchas ventajas en proyectos colaborativos especialmente cuando se trabaja con equipos multidisciplinarios o cuando los requerimientos cambian con frecuencia. Segun el documento, Scrum promueve ciclos cortos de desarrollo.

10

Desarrollo e implementación de una aplicación web para la gestión de inventarios y ventas de la Lubrificadora lubricante freddy

En este artículo se presentan problemáticas significativas debidas al manejo manual y desorganizado de sus procesos, generando un inventario impreciso y dificultades en el control de las transacciones comerciales. La solución tecnológica propuesta permite a los usuarios llevar el control preciso y en tiempo real de todos los productos almacenados, facilitando el registro meticuloso de entradas y salidas de monedaria, incluyendo el cálculo automático de costos mediante el método de promedio ponderado en el Kardex, así como la gestión agil y confiable de las ventas a través de un módulo de facturación electrónica integrado con los estandares de SRI. Adicionalmente la aplicación genera reportes analíticos detallados que sirven como base sólida para la toma de decisiones estratégicas mejorando la eficiencia operativa y la rentabilidad del negocio. Para garantizar una ejecución ordenada y eficaz del proyecto se adoptó la metodología ágil SCRUM la cual sustentada en el trabajo en equipo iterativo y la organización rigurosa de tareas que administrada mediante la herramienta JIRA en su versión gratuita permitiendo una gestión metódica de todas las actividades, la priorización efectiva de los requisitos y un avance constante y medible en el desarrollo del sistema web el cual fue construido utilizando un stack tecnológico moderno que incluye Spring Boot con security JWT para el backend, Vue.js para el frontend y PostgreSQL como sistema de gestión de base de datos.

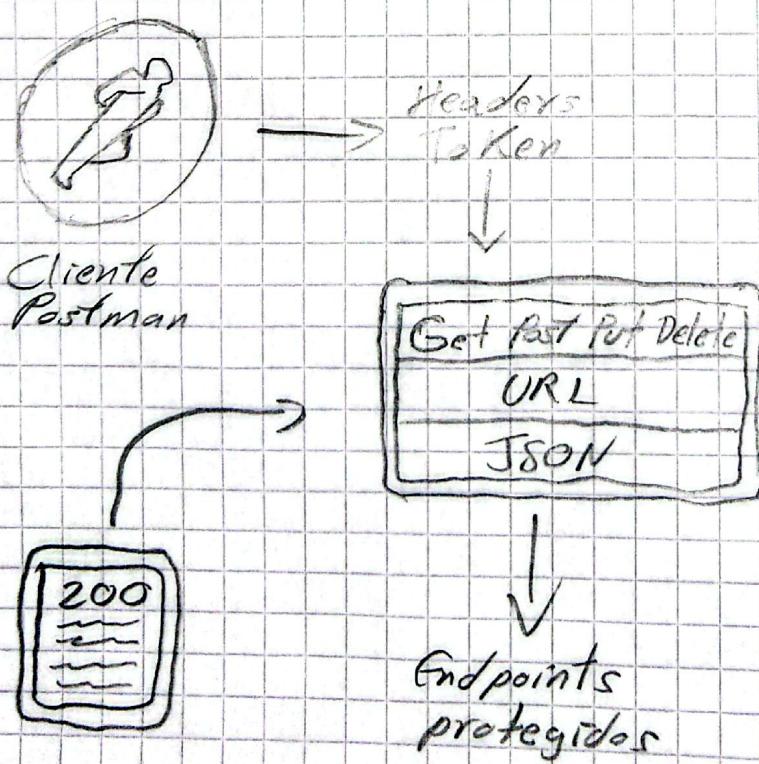


Reflexion:

La implementacion de JWT en Spring Boot es mucho mas que una solucion tecnica para autenticacion, segun el documento JWT permite generar tokens seguros que encapsulan la identidad del usuario, sus roles y permisos ademas de que pueda ser verificados sin necesidad de mantener sesiones en el servidor. JWT es muy bueno para la seguridad y por eso se implementa en mi proyecto.

II APIs RESTFUL PARA INTEROPERABILIDAD ENTRE APLICACION MOVIL Y APLICACION WEB

Se aborda la necesidad de integrar eficazmente plataformas digitales para fortalecer la promoción de productos artesanales en Tabasco mediante el diseño de una solución tecnológica que conecta de forma segura y eficiente la aplicación móvil y web existente del IFAT (Instituto para el Fomento de las Artesanías del Tabasco) a través de APIs RESTful. La metodología incluye el uso de arquitecturas escalables, microservicios y protocolos de autenticación JWT para garantizar la seguridad y flexibilidad del sistema. Como resultado se desarrolla una infraestructura que permite la comunicación eficiente y en tiempo real entre la aplicación móvil y la web, facilitando el acceso a la información clave como productos, puntos de venta, inventario y usuarios. Se concluye que la interoperabilidad tecnológica basada en APIs RESTful es una herramienta clave para el desarrollo cultural y comercial en entornos digitales, a pesar de los desafíos técnicos y contextuales identificados como la complejidad en el mantenimiento y la limitada conectividad en algunas áreas. El proyecto sienta las bases para futuras mejoras incluyendo APIs Gateway y herramientas de análisis para optimizar el ecosistema digital del IFAT.



ReDoc2

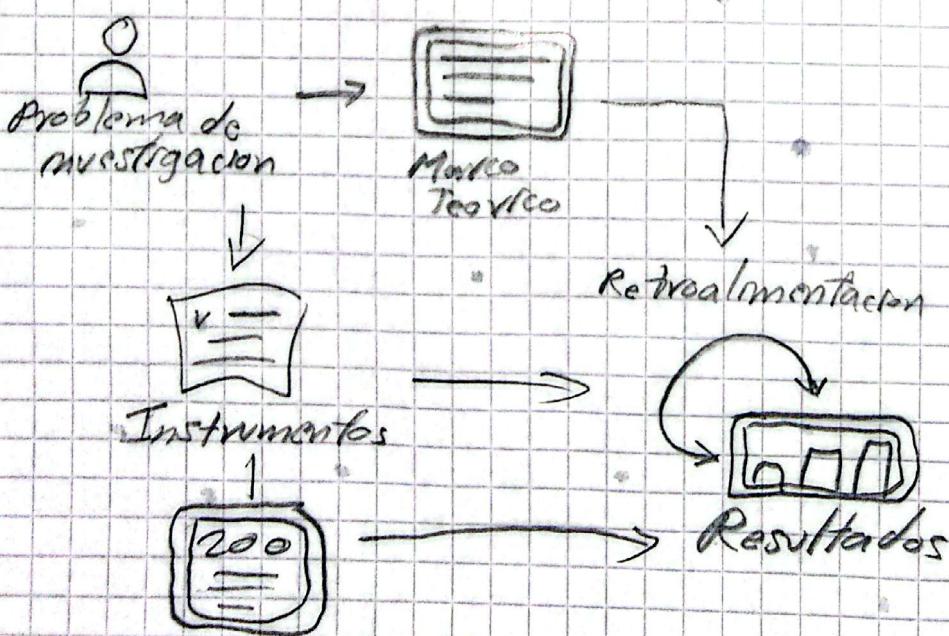
Postman es una herramienta para probar o simular peticiones http, validar respuestas para documentarlas, no digo que sea mala al contrario es muy buena pero hay herramientas que automatizan esto. Ejemplo Swagger que ya directamente automatiza la documentación de Endpoint automáticamente y ya eso depende de cada quien que herramienta utilizar.

12

Desarrollo de frontend para es sistema de gestion de pacientes de la Unidad de diagnostic, investigación Psicopedagógica y apoyo a la inclusión (UDIPSAI)

Presenta el desarrollo de una aplicación web frontend para modernizar la gestión de pacientes en la Unidad de diagnóstico, investigación Psicopedagógica y apoyo a la inclusión de la universidad católica de Cochabamba. El sistema fue diseñado para reemplazar el manejo análogo de información que dificultaba el acceso a los datos, incrementa su vulnerabilidad y contribuye a la revictimización de pacientes mediante una solución digital segura y eficiente. El desarrollo se guio por la metodología ágil SCRUM, permitiendo un proceso iterativo e incremental que incluyó la recolección de requisitos con el estadio FURPS+, el diseño de prototipos en figuras y la implementación con tecnologías modernas como React.js, Next.js, Zustand y Ant Design (antd). Se priorizo el diseño centrado en el usuario y la accesibilidad cumpliendo con las pautas WCAG. El sistema incluye funcionalidades como registro, y gestión de pacientes, generación de fichas especializadas, seguimiento de citas, historial de cambios y generación de informes en PDFs. La evaluación post-implementación mostró una alta satisfacción entre los usuarios, el 100% calificó la aplicación entre 4 y 5 en la escala de Likert, y el 66.6% reportó una reducción en el tiempo requerido para completar registros. El proyecto demuestra como la combinación de metodologías ágiles, tecnologías modernas y un enfoque centrado en el usuario puede optimizar procesos críticos en entornos de salud y educación especial.

Estructura Metodológica

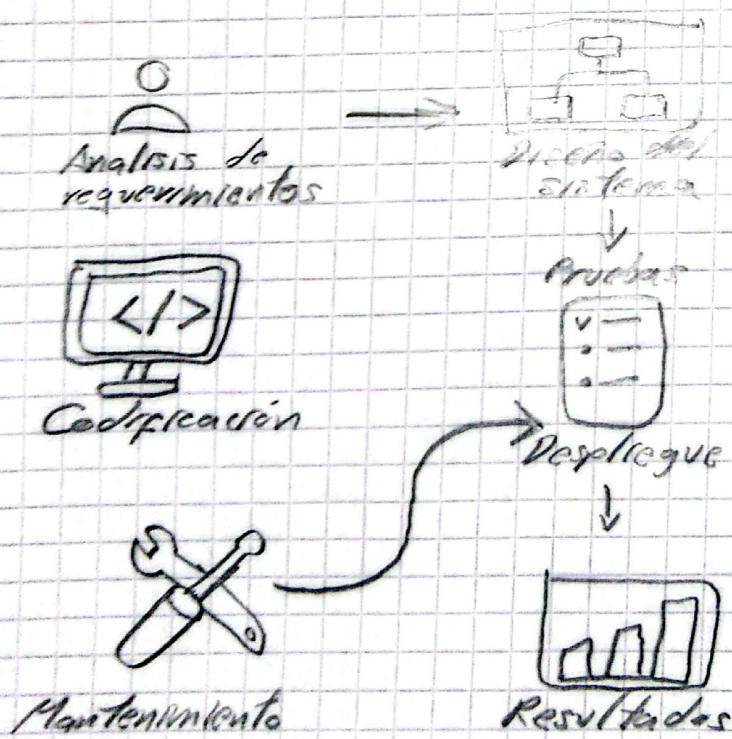


Reflexión:

Scrum es una metodología que se utiliza en mi proyecto con la delegación de responsabilidades y esto si hizo en mi proyecto se hacen dailys es decir sesiones de unas 10 a 20 minutos donde cada quien dice que hizo que se va a hacer y etc... Scrum me parece una metodología muy buena por lo mismo antes dictó además de mantener una estructura de personal responsable como el stakeholder.

Construcción de Software

Es un programa de estudio integral y práctico diseñado para introducir a los estudiantes en el desarrollo moderno de software mediante un enfoque ágil, colaborativo y basado en proyectos reales. A lo largo de sus cuatro unidades seccionales, los alumnos avanzan desde conceptos fundamentales como el desarrollo rápido de aplicaciones en Python, el uso de inteligencia artificial generativa para asistencia en código y la aplicación de principios de código limpia y manejo de excepciones hasta técnicas avanzadas de control de versiones con Git y Github, incluyendo flujos de trabajo colaborativos, resolución de conflictos y automatización de procesos CI/CD con Github Actions. La tercera unidad se centra en garantizar la calidad del software mediante pruebas unitarias, desarrollo guiado por pruebas (TDD) y el uso de ORMs para la gestión de base de datos, mientras que la cuarta unidad culmina el proceso con el diseño de interfaces gráficas usables y accesibles y el despliegue portable de aplicaciones utilizando Docker y Docker Compose. El eje central de la guía es un proyecto integrador (como una conversión de monedas) que sintetiza todos los conocimientos adquiridos, comentando el trabajo en equipo con roles definidos bajo metodología SCRUM, el uso de herramientas de desarrollo estándar de la industria y la entrega iterativa de software funcional, bien documentado y preparado para entornos productivos.

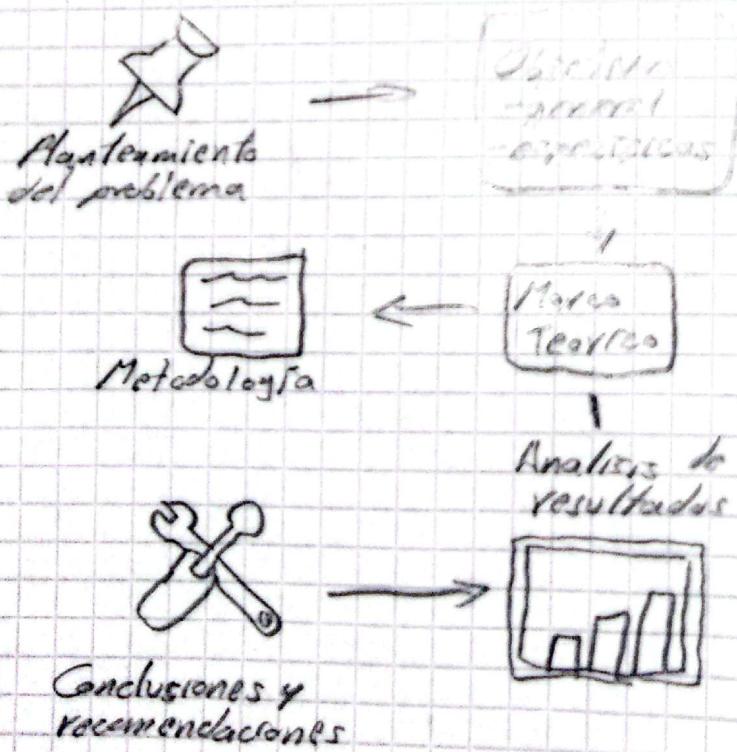


Reflexión:

Se puede ver una guía práctica y estructurada para la construcción de software en contexto educativo dandoleas Metodologías y trabajo colaborativo, buenas prácticas este documento recuerda que a veces los proyectos no sean demasiados complejos la disciplina en el proceso es lo que marca la diferencia ya que siempre se necesita un flujo de trabajo que garantice (exportar datos impecablemente) que cada cambio este validado, documentado y aprobado.

Practicante en lenguaje de programación Java y nuevas tecnologías.

Se desarrollaron y fortalecieron competencias clave en el ámbito del desarrollo de software moderno trabajando dentro del área de sistema de la empresa bajo la metodología SCRUM, participaron activamente en la optimización y refactorización de microservicios desarrolladas en java con spring boot, modifíco APIs existentes para mejorar su integración con el frontend construido en Angular e implemento nuevas funcionalidades en la plataforma EasyHub como validaciones de estado de programa y bloqueo condicional de campos en formularios. Utilizo bases de datos SQL y NoSQL se aplicaron pruebas unitarias y de integración o JUNIT para garantizar la calidad del código y gestiono el control de versiones mediante Git y SourceTree. Esta experiencia no solo lo permitió consolidar y expandir sus conocimientos técnicos en desarrollo full-stack ya arquitecturas de software sino también desarrolla habilidades blandas esenciales como el trabajo en equipo, la comunicación efectiva y la adaptación a un entorno empresarial dinámico contribuyendo significativamente a la mejora y escalabilidad de los sistemas de la empresa mientras se preparaba para futuros desafíos en el campo de la ingeniería de software.

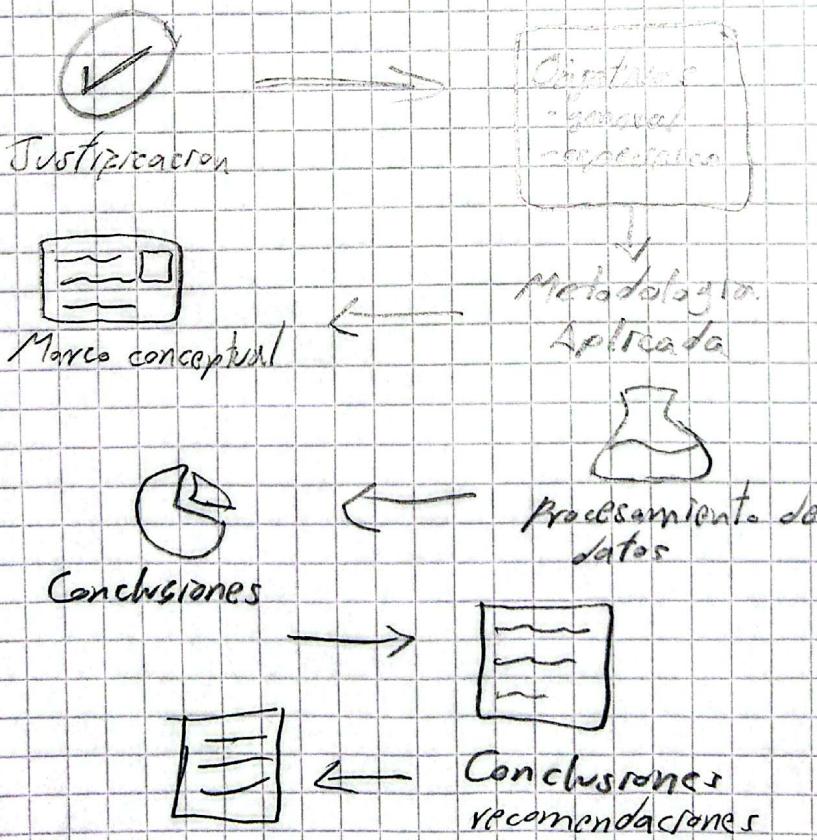


Reflexión:

Me queda claro que la metodología y las buenas prácticas no son un adorno sino el esqueleto que sostiene un proyecto vivo, se ve como la disciplina técnica (pruebas o control de versiones) y la disciplina humana (comunicación o adaptación) se entrelazan para que el resultado no sea solo "código que funciona" sino una solución sostenible y mejorable.

15 Estimapro: sistema de información para estimaciones de propuestas de desarrollo de software para la empresa SEED EM.

El proyecto estimapro fue desarrollado como un sistema de información web para abordar la problemática de demoras en la empresa Seed EM. el cual se realizaba de forma manual mediante plantillas de excel. la solución tecnológica implementada con una arquitectura basada en React para el frontend, Django para el backend y Neo4j como base de datos de grafos. automatizando tareas clave como la gestión, búsqueda y revisificación de HV así como la generación automática del documento de propuestas. Durante una prueba piloto se logró una reducción del 33,6% en el tiempo promedio de elaboración de estimaciones para proyectos de pequeña escala, mejorando significativamente la capacidad de respuesta comercial y fomentando una transformación cultural hacia la eficiencia y colaboración interna. Además el sistema tiene las bases técnicas para futuras evoluciones, como la incorporación de inteligencia artificial y grafos de conocimientos.

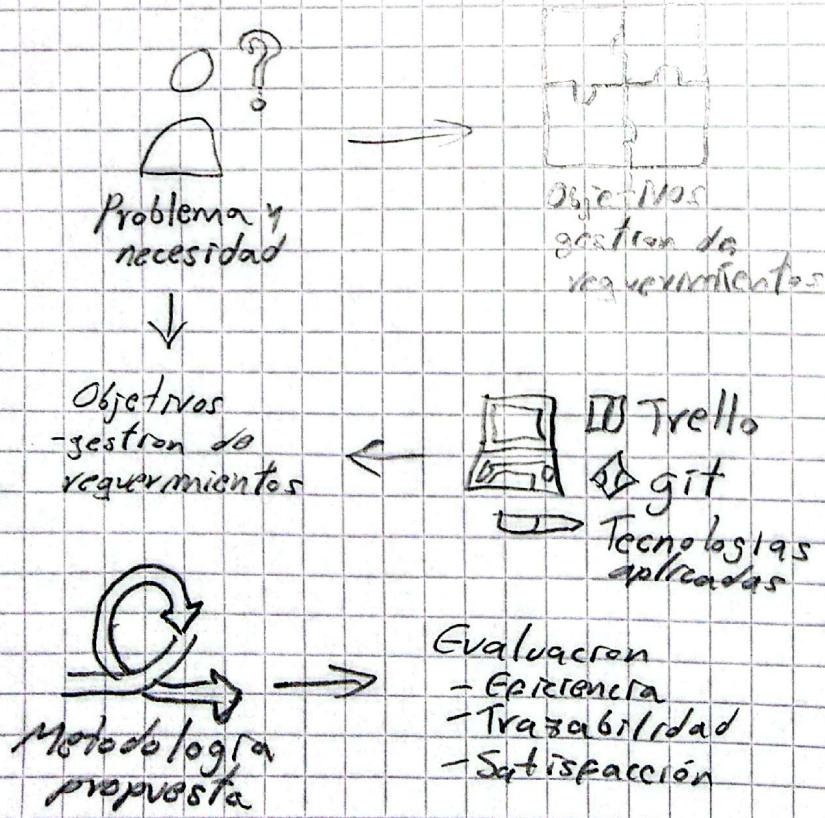


Reflexión:

Este documento me habla de un sistema bien diseñado no solo mejores procesos si no que transforma culturas. Se puede ver referencia metodológica, arquitectónica y humana no sirve para saber que el Backend no es solo lógica y datos sino también median entrar personas, procesos y propósitos.

Caracterización de especificaciones de requerimientos en entornos Ágiles : Historias de Usuarios

Este documento de investigación está realizado en el marco de la especialización de ingeniería en sistemas de información de la Universidad Tecnológica Nacional, facultad regional Córdoba, ofreciendo un análisis exhaustivo de las prácticas de especificación de requerimientos dentro de metodologías ágiles con especial foco en Scrum y el uso de Historias de usuarios. El trabajo contrasta este enfoque ágil con los métodos tradicionales de especificación basado en el estándar IEEE 830, examinando en profundidad a los cambios y su capacidad para ser comprendidas tanto por clientes como desarrolladores así como sus posibles desventajas en proyectos de gran escala. Asimismo se detalla el modelo INVEST como guía para la redacción de historias de usuarios efectivas y se concluye que lejos de ser excluyentes los requerimientos tradicionales y las historias de usuarios pueden coexistir y complementarse, combinando la precisión y gestionabilidad de los primeros con la flexibilidad y orientación al valor de los segundos sentando así las bases para futuras investigaciones sobre trazabilidad en entornos ágiles complejos.

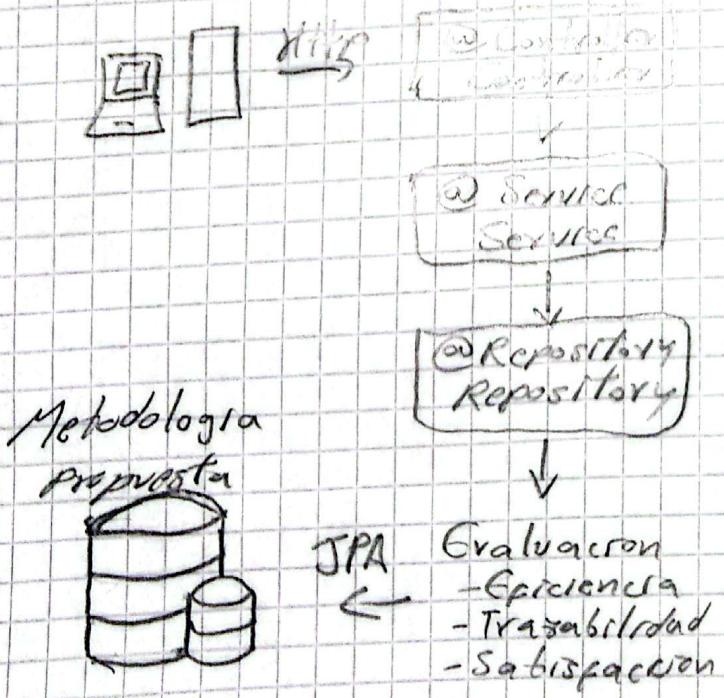


Reflexión:

En este documento se habla que los requerimientos no son solo técnicos sino que también son expresiones de necesidades humanas. Esto nos recuerda que cualquier requerimiento mal establecido puede hacer que nuestro proyecto en código sea bueno pero inútil en la vida real.

17 Desarrollo de una aplicación web para E-commerce utilizando Spring Framework

presentando en el Instituto Tecnológico de Tecomatlán para obtener el grado de Ingeniero en Sistemas Computacionales. Documenta el diseño, desarrollo e implementación de una plataforma de comercio electrónico para una tienda especializada en materiales didácticos y regalos de Mesa de la Ciudad de México. El estudio se enfoca en la aplicación de metodologías ágiles, específicamente Extreme Programming (XP) para garantizar flexibilidad, adaptabilidad y una estrecha colaboración con el cliente durante el ciclo de desarrollo. Tecnológicamente el proyecto se sustenta en el ecosistema Spring Framework (Spring Boot, Spring Data JPA, Spring Web, Spring Security) para el backend utilizando PostgreSQL como sistema de gestión de base de datos y Maven para la gestión de dependencias. Asimismo, se destaca el uso de herramientas modernas de desarrollo como IntelliJ IDEA y Visual Studio Code. El marco teórico abarca desde los fundamentos del e-commerce y las aplicaciones web hasta los modelos de bases de datos relacionales y principios de programación clave como JAVA y TypeScript. El trabajo concluye con la entrega de una aplicación robusta, escalable y segura que facilita la expansión del negocio mejorando la gestión de inventario y optimizando la experiencia de usuario, sintiendo las bases para futuras mejoras e integraciones.

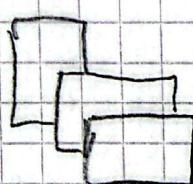


Reflexión:

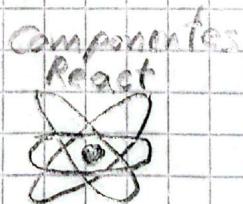
La arquitectura no es solo diseño técnico sino pedagógica,
es decir para que otras personas puedan entender.
es decir que se construye con propósito con claridad para
dejar un legado porque un sistema bien hecho no solo funciona
sino que se entiende se reproduce y se mejora.

Desarrollo de un frontend en React JS

Trabajo final de grado presentado en la Universidad Javine I por Pablo Berdel Marin documenta el diseño, implementación y despliegue del frontend de constructos una aplicación web, destinada a la gestión de certificaciones profesionales en el sector de la construcción para una organización holandesa. El proyecto se enmarca en un entorno de desarrollo dirigido mediante Scrum, con una estrecha colaboración entre los equipos locales y remotos de la empresa E-force. Tecnológicamente, se fundamenta en React JS para la construcción de interfaces modulares y reactivas, utilizando Bootstrap para el diseño responsive y herramientas como Postman para el testing de APIs RESTful desarrolladas en .NET. Crea la memoria detallada desde el análisis de requisitos y la planificación temporal mediante diagramas de Gantt hasta la implementación de componentes jerárquicos, la gestión de estado mediante programación funcional y el control de acceso por roles de usuarios. Asimismo se abordan temas como la migración de una base de datos MySQL a SQL Server y la interacción continua mediante Bitbucket. El trabajo concluye con la entrega de una interfaz funcional usable y bien estructurada, que cumple con las objetivas de mejorar la experiencia de usuario respecto al sistema anterior y sirve las bases para futuras iteraciones en un entorno productivo real.



Estado y Props



Consumo de APIs

fetch
axios

useState
useEffect
useContext

• Vite
• Webpack
• Babel



React Router

Stilos

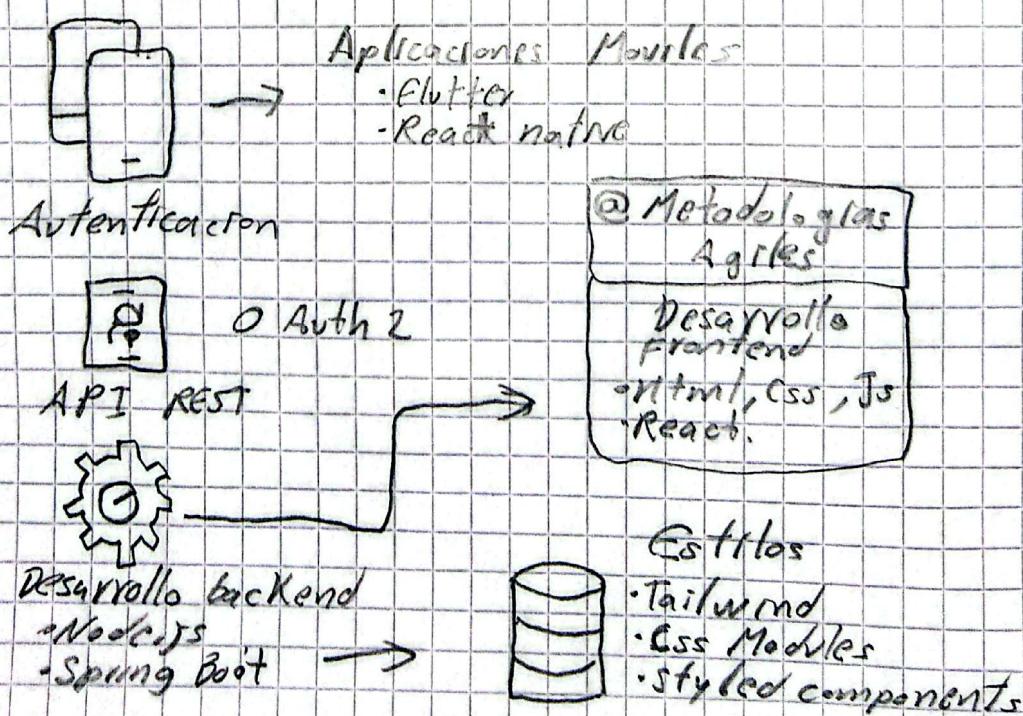
- Tailwind
- CSS Modules
- Styled Components

Reflexión:

El frontend no es solo estética sino arquitectura visible. Cada componente, cada llamada de API es una decisión que afecta como las personas entienden, usan y comparten en el sistema. Se subraya entendiendo que el controller no es el fin del desarrollo ya que con el frontend se complementa con el Backend porque al final se está construyendo una experiencia.

Desarrollo de dispositivos móviles y full stack

Presentando un proyecto en la universidad de la laguna donde se documenta el diseño, implementación y despliegue del FundMyMusic una plataforma de crowdfunding musical que conecta a artistas emergentes con su público mediante la venta anticipada de entradas para conciertos. El proyecto adopta un enfoque full stack, integrando un backend desarrollado en Node.js y express que sirve una API RESTful para base de datos NoSQL con MongoDB para gestionar usuarios, conciertos y transacciones en frontend web en React Native para android que permite a los usuarios descubrir, guardar y comprar entradas. Se incorporan servicios externos como Cloudinary para la gestión de imágenes y nodemailer para la confirmación de cuentas y recuperación de contraseñas. El despliegue se automatiza mediante Github Actions, alojando el backend en Heroku, el frontend web en Iemos con dominio personalizado y distribuyendo la aplicación móvil mediante descarga directa de APK. El trabajo concluye con una solución funcional y escalable que cumple con el objetivo de facilitar la financiación de conciertos musicales y se proponen líneas futuras como la integración de encuestas de pago, sistema de valoración y análisis estadísticos, reflejando tanto el dominio técnico de las tecnologías empleadas como una compresión profunda de las necesidades del sector musical emergente.

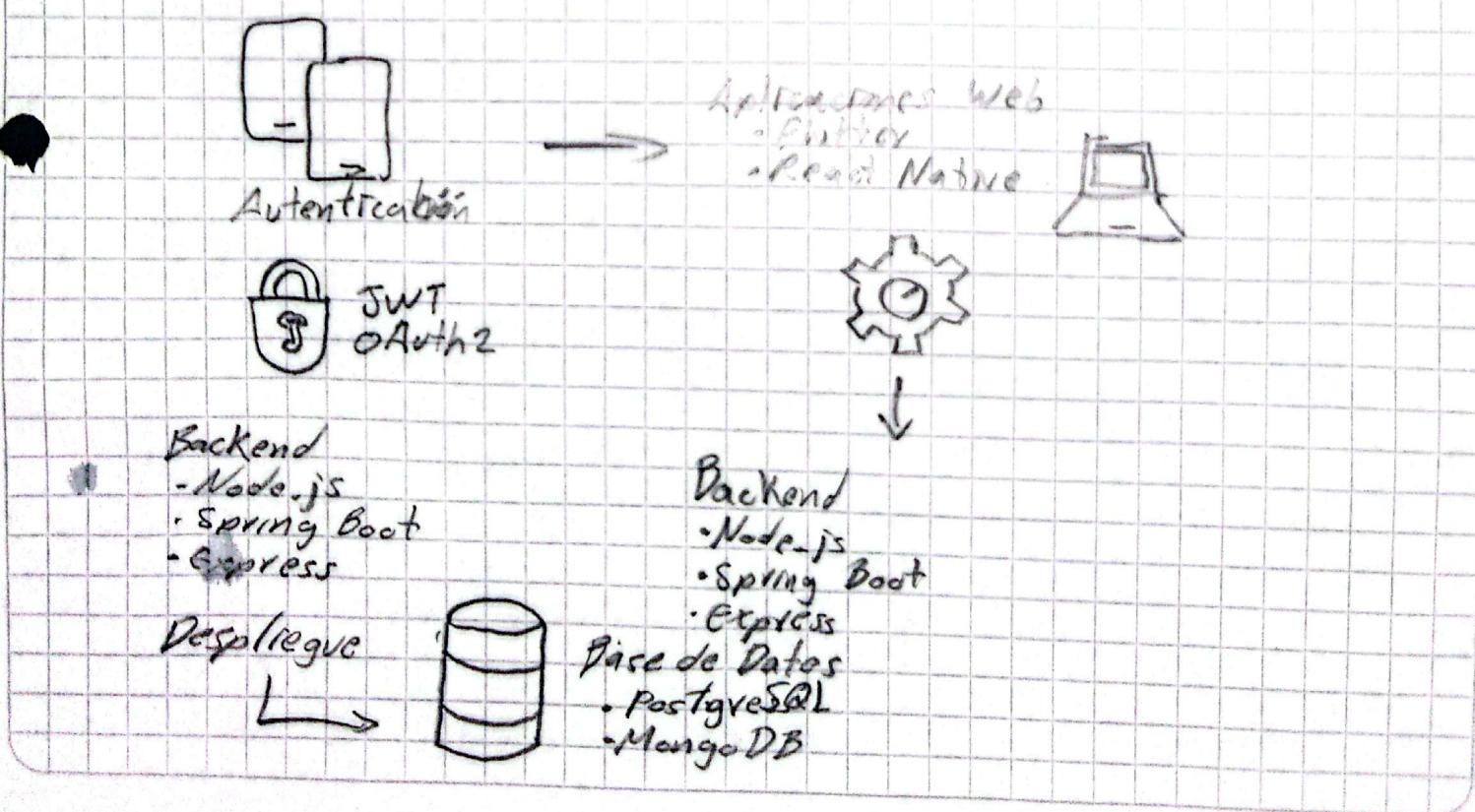


Reflexión:

Este documento no es más que uno de muchos proyectos enfocados más en el front-end más que nada en la experiencia de usuario pero enfocada en un framework para móvil, hace recordar que el trabajo a proyecto se debe pensar en como se consume como se entiende y como se siente.

Analisis de frameworks para desarrollo de aplicaciones móviles y web.

Se documenta el diseño implementación y evaluación comparativa de una aplicación tipo todo desarrollada con tres frameworks Multiplataformas: React Native, Ionic y Flutter. El proyecto adopta un enfoque práctico y analítico, integrando un backend único desarrollado con Node.js y MongoDB que sirve una API RESTful para gestionar tareas, imágenes, fechas y ubicaciones junto con servicios de Firebase, Cloud Functions y Cloud Messaging para el envío automatizado de notificaciones push. Se evalúan aspectos como la facilidad de desarrollo, el rendimiento, la estructura de proyectos y la simplicidad con aplicaciones nativas. El trabajo concluye que Flutter destaca por su alto rendimiento, baja curva de aprendizaje y arquitectura basada en widgets, seguido de React Native como alternativa sólida para desarrolladores con experiencia en React, mientras que Ionic muestra limitaciones en rendimiento y escalabilidad. Se proporcionan todos los repositorios de código abierto para su consulta y replicación y se proponen líneas futuras como la adopción de arquitecturas avanzadas (Bloc), la inclusión de nuevos frameworks como Ionic React y la ampliación de funcionalidades para evaluar escalabilidad. El estudio refleja tanto el dominio teórico de las tecnologías evaluadas como una comprensión crítica de las ventajas y desventajas de cada framework en el contexto del desarrollo móvil y web.



Reflexión:

Se puede observar comparación técnica entre flutter, React Native e Ionic. Es para el desarrollo móvil, reducir errores de entrada y hacer una sola idea pueda llegar a millones de usuarios sin duplicar esfuerzos es como la búsqueda por eficiencia sin sacrificar experiencia. Pero más allá del rendimiento o la curva de aprendizaje lo que realmente importa es el propósito detrás del código.