

Meta información del TFG

Índice

Ejemplo

Este índice se basa en el flujo:

- Introducción
- Tecnologías y herramientas
- Metodología
- Análisis
- Modelado y diseño
- Desarrollo e implementación
- Conclusiones y líneas futuras

QRest

Lluvia de ideas Ideas:

- Podemos usar el índice anterior y dentro de cada apartado centrarnos en los puntos fuertes.
- Dedicar una sección para explicar en detalle cada punto fuerte y hacer referencia a la sección que explica cada uno.
- Hacer el índice basado en los casos de uso, partes de la aplicación o en los sprints.

Conclusión: Tras releer el email de carlos y pensar sobre toda la lluvia de ideas, lo más coherente y lo que minimiza más los riesgos es seguir el índice de ejemplo pero añadir una sección para destacar y detallar los puntos fuertes. Eso sí, no pisando contenido de otras secciones.

Índice resultado Este índice se basa en los puntos fuertes:

- Introducción (Puntos fuertes)
 - Problemas del sector
 - Análisis de competidores
 - Solución buscada (Requisitos)
 - Puntos fuertes con respecto a competidores
- Puntos fuertes
 - Producto
 - Ingeniería
 - Diseñado pero no implementado
- Tecnologías y Herramientas
- Metodología
- Análisis
- Modelado y diseño
- Desarrollo e implementación
- Conclusiones y líneas futuras

Separación de responsabilidades:

En el apartado de Puntos fuertes se hablará más del por qué y del cómo pero sin entrar en detalles técnicos, ya que los detalles técnicos se abordarán en sus respectivos apartados. A los cuales se harán referencia desde el apartado de Puntos fuertes.

Puntos fuertes

Diseñado pero no implementado

- Solución segura y cómoda de generación de pedidos con QR.
- Flujo flexible de gestión de comandas del restaurante.
- Conexión con la impresora de tickets con librería de JS.
- Búsquedas en la carta, con filtros.
- Frontend con elementos complejos.
- Frontend para modificar la carta.
- Llamar al camarero desde la app.
- Sincronización de estado de servidores.

Producto

- Privacidad de datos de los clientes con localStorage.
- Pagos por separado.
- Recomendaciones de repetir lo pedido anteriormente y etiquetas.
- Confirmación de pago.
- Sincronización.
- Flexibilidad.
- Seguridad de API.
- Mostrar alérgenos.

Ingeniería

- Arquitectura en capas al estilo clean Achitecture.
- Tests de integración.
- Estructura de datos capaz de representar toda la casuística de restaurantes.
- Transacciones seguras de casos de uso.
- Metodología ágil basada en sprints.