**Requerimiento Funcionales en un Proyecto**

1. **Análisis de requisitos**: Comienza por comprender completamente los requerimientos funcionales. ¿Qué funcionalidades debe tener tu proyecto? ¿Cuáles son las expectativas del usuario? Documenta todo de manera clara.
2. **Diseño y arquitectura**: Define la arquitectura de tu proyecto. ¿Qué tecnologías utilizarás? ¿Cómo se relacionarán los diferentes componentes? Esto te ayudará a visualizar cómo implementar los requerimientos.
3. **Desarrollo iterativo**: Divide el proyecto en tareas más pequeñas y trabaja en ellas de manera iterativa. Implementa una funcionalidad a la vez y verifica que cumpla con los requerimientos.
4. **Codificación**: Escribe el código siguiendo las mejores prácticas de programación. Utiliza el lenguaje de programación que mejor se adapte a tus necesidades (por ejemplo, JavaScript o PL/SQL).
5. **Pruebas unitarias**: A medida que implementas cada funcionalidad, realiza pruebas unitarias para asegurarte de que funcione correctamente. Corrige cualquier error que encuentres.
6. **Integración y pruebas de sistema**: Integra todas las partes del proyecto y realiza pruebas de sistema. Verifica que todas las funcionalidades interactúen correctamente.
7. **Documentación**: Documenta tu código y las decisiones de diseño. Esto facilitará el mantenimiento y futuras actualizaciones.
8. **Despliegue**: Finalmente, despliega tu proyecto en un entorno real. Si buscas opciones de alojamiento gratuito para PL/SQL, considera servicios como Oracle Cloud Free Tier o Heroku.

**¿Cómo puedo aplicarlo para un proyecto real?**

1. **Comprende el dominio**: Antes de comenzar, sumérgete en el dominio del proyecto. ¿Qué problema estás resolviendo? ¿Quiénes son los usuarios finales? Esto te ayudará a definir los requerimientos funcionales con mayor precisión.
2. **Planificación y estimación**: Divide el proyecto en tareas más pequeñas y estima el tiempo necesario para cada una. Prioriza las funcionalidades según su importancia y dependencias.
3. **Diseño de la base de datos**: Si tu proyecto implica almacenamiento de datos, diseña la estructura de la base de datos. Define tablas, relaciones y restricciones.
4. **Desarrollo iterativo**: Implementa las funcionalidades una por una. Sigue las mejores prácticas de codificación y documenta tu trabajo.
5. **Pruebas exhaustivas**: Realiza pruebas unitarias, de integración y de sistema. Asegúrate de que cada funcionalidad cumpla con los requerimientos y no cause efectos secundarios.
6. **Documentación continua**: Documenta el código, decisiones de diseño y configuración del proyecto. Esto facilitará el mantenimiento y la colaboración con otros desarrolladores.
7. **Despliegue y monitoreo**: Despliega el proyecto en un entorno real. Configura herramientas de monitoreo para detectar problemas y mejorar el rendimiento.
8. **Itera y mejora**: Obtén retroalimentación de los usuarios y sigue mejorando el proyecto. Ajusta los requerimientos según sea necesario.

**¿Cuáles son los** **tipos de documentación que se utilizan en la programación?**

1. **Documentación interna**:

* Documentación de requisitos: Describe qué debe hacer el software y cómo debe hacerlo. Incluye información sobre las funciones que el software debe proporcionar, los datos que debe almacenar y los usuarios que lo utilizarán.
* **Documentación** **de diseño**: Describe cómo se implementarán los requisitos del software. Incluye información sobre la arquitectura del software, el diseño de las clases y los algoritmos que se utilizarán.
* **Documentación de código:** Explica cómo está construido el software. Incluye comentarios que explican el propósito de cada bloque de código y diagramas que ilustran la estructura de los archivos.

1. **Documentación externa**:

* **Manuales de usuario**: Explican cómo utilizar el software. Incluyen información sobre cómo iniciarlo, cómo utilizar sus funciones y cómo resolver posibles dudas.
* **Documentación de API**: Proporciona información sobre cómo utilizar las funciones de la API para integrar el software con otros sistemas.
* **Documentación de marketing**: Utilizada para promocionar el software. [Incluye detalles sobre características, beneficios y cómo adquirir el software](https://hermescoder.com/tipos-de-documentacion-en-programacion.html).

**Pruebas unitarias en la programación**

Es un bloque de código que verifica la precisión de un bloque más pequeño y aislado de código de aplicación, normalmente una función o un método, la prueba unitaria esta diseñada para verificar que el bloque de código se ejecuta según lo esperado, de acuerdo con la lógica teórica del desarrollador

**Documentación Técnica y documentación de software**

**Documentación técnica:**

* **Comentarios en el código**: Explican por qué se ha tomado una decisión en particular, no solo qué hace el código.
* **Guías de instalación**: Detallan cómo instalar y configurar el software.
* **Manuales de usuario**: Proporcionan instrucciones claras sobre cómo utilizar el software.
* [**Especificaciones de la API:** Describen cómo interactuar con las funciones del software](https://bing.com/search?q=Documentacion+Tecnica+y+documentacion+de+software).

**Características de una buena documentación:**

* **Claridad y concisión**: Evita la jerga innecesaria y comunica de manera directa.
* **Enfoque en el usuario**: Piensa en quienes leerán la documentación y cómo les ayudará.
* [**Actualización constante**: Mantén la documentación al día a medida que evoluciona el software](https://www.arsys.es/blog/hacer-documentacion-tecnica-software).

**¿Cómo analizar documentación técnica?**

1. **Definir la pregunta de investigación:**

* [Antes de comenzar, formula una pregunta específica que deseas responder mediante el análisis de los documentos](https://bing.com/search?q=C%c3%b3mo+analizar+documentaci%c3%b3n+t%c3%a9cnica).
* Por ejemplo, podrías preguntarte: “¿Cómo se implementa la autenticación en este sistema?”.

1. **Seleccionar los documentos pertinentes:**

* Identifica los documentos relevantes para tu investigación. [Pueden ser manuales, especificaciones, informes técnicos o cualquier otro recurso relacionado con el software](https://bing.com/search?q=C%c3%b3mo+analizar+documentaci%c3%b3n+t%c3%a9cnica).
* Considera la calidad, la fuente y la actualidad de los documentos.

1. **Preparar y organizar los documentos:**

* Organiza los documentos de manera accesible. Puedes utilizar herramientas como carpetas digitales o etiquetas para clasificarlos.
* Realiza copias de los documentos para tomar notas sin alterar los originales.

1. **Revisión inicial y codificación:**

* Realiza una revisión rápida de los documentos para obtener una visión general.
* Codifica la información relevante utilizando etiquetas o categorías. [Por ejemplo, puedes etiquetar secciones sobre “seguridad” o "arquitectura"](https://bing.com/search?q=C%c3%b3mo+analizar+documentaci%c3%b3n+t%c3%a9cnica)

1. **Análisis e interpretación de datos:**

* Profundiza en los documentos. Lee detenidamente las partes relevantes.
* Compara y contrasta la información entre diferentes documentos.
* Busca patrones, tendencias o discrepancias.

1. **Presentar resultados y extraer conclusiones:**

* Resume tus hallazgos en un informe o presentación.
* [Extrae conclusiones basadas en la evidencia documental](https://bing.com/search?q=C%c3%b3mo+analizar+documentaci%c3%b3n+t%c3%a9cnica).

**documentación técnica de un proyecto real**

**McDonald’s**

**Sistemas Informáticos**

1. **Punto de venta (POS)**:

* McDonald's utiliza sistemas de punto de venta avanzados en todos sus restaurantes. Estos sistemas no solo registran pedidos, sino que también gestionan inventarios, controlan la calidad del servicio y recopilan datos para análisis.
* La marca McDonald's utiliza el software NewPOS, que se integró en la marca para optimizar los sistemas y aumentar la rentabilidad.

1. **Sistemas de gestión de pedidos y entrega**:

* McDonald's ha invertido en sistemas de gestión de pedidos para mejorar la precisión y eficiencia de la entrega de pedidos a través de aplicaciones móviles y kioscos de autoservicio en los restaurantes.

1. **Gestión de cadena de suministro**:

* Utilizan sistemas integrados para la gestión de la cadena de suministro, que incluyen el seguimiento de inventarios en tiempo real, la planificación de la demanda y la coordinación logística para garantizar la entrega oportuna de ingredientes frescos a los restaurantes.

1. **Analítica y Business Inteligencie**:

* McDonald's emplea herramientas avanzadas de analítica y Business Inteligencie para analizar datos operativos, de ventas y de clientes. Esto les permite tomar decisiones informadas sobre precios, promociones, gestión de inventarios y expansión de negocios.

1. **Seguridad y protección de datos**:

* Dada la sensibilidad de la información de clientes y transacciones financieras, McDonald's implementa estrictas medidas de seguridad cibernética y protección de datos para garantizar la privacidad y la integridad de la información.

1. **Tecnologías emergentes**:

* La compañía ha estado explorando tecnologías emergentes como la inteligencia artificial y el aprendizaje automático para mejorar la personalización del servicio al cliente, optimizar la eficiencia operativa y predecir la demanda de productos.

1. **Infraestructura IT y redes**:

* McDonald's gestiona una infraestructura de IT robusta que incluye redes de comunicación seguras y sistemas de almacenamiento de datos escalables para soportar sus operaciones globales.