# Investigación II



Aprendiz

Sergio David Rodriguez Robayo

Servicio nacional de aprendizaje (SENA) – Centro de electricidad, electrónica y telecomunicaciones (CEET)

Instructora Isaura María Suarez Novoa

Bogotá D.C

2023

## ¿Qué es la arquitectura de datos y en que radica su importancia?

Manejan el almacenamiento la organización e integración de datos lo que su importancia radique en los datos que son el activo crucial para organizaciones ya que mejoran la toma de decisiones y realizar otros procesos para generar oportunidades de negocio.

### Menciona y explica cada uno de los métodos y procesos propuestos para diseño de proyectos de análisis y ciencia de datos.

Los métodos y procesos propuestos para el diseño de proyectos de análisis y ciencia de datos se pueden dividir en dos grandes categorías:

Metodologías tradicionales: Estas metodologías se basan en los principios del desarrollo de software tradicional y se centran en el ciclo de vida del proyecto.

Metodologías ágiles: Estas metodologías se centran en la flexibilidad y la adaptación a los cambios, y se basan en principios como la iteración, la colaboración y la comunicación.

Metodologías tradicionales

Las metodologías tradicionales más utilizadas para el diseño de proyectos de análisis y ciencia de datos son:

El modelo en cascada: Este modelo es el más tradicional y consta de una serie de fases secuenciales que se completan una tras otra. Las fases del modelo en cascada son:

Requisitos: En esta fase se identifican los requisitos del proyecto.

Diseño: En esta fase se diseña la solución a los requisitos del proyecto.

Implementación: En esta fase se implementa la solución diseñada.

Pruebas: En esta fase se prueba la solución implementada.

Implementación: En esta fase se implementa la solución probada.

El modelo en espiral: Este modelo es una evolución del modelo en cascada y permite un mayor grado de iteración. El modelo en espiral consta de las siguientes fases:

Planificación: En esta fase se planifica el proyecto y se identifican los riesgos.

Especificación: En esta fase se especifican los requisitos del proyecto.

Diseño: En esta fase se diseña la solución a los requisitos del proyecto.

Implementación: En esta fase se implementa la solución diseñada.

Pruebas: En esta fase se prueba la solución implementada.

Evaluación: En esta fase se evalúa la solución implementada y se identifican los riesgos para la siguiente iteración.

El modelo V: Este modelo combina los principios del modelo en cascada con los principios del modelo en espiral. El modelo V consta de las siguientes fases:

Requisitos: En esta fase se identifican los requisitos del proyecto.

Diseño: En esta fase se diseña la solución a los requisitos del proyecto.

Implementación: En esta fase se implementa la solución diseñada.

Pruebas: En esta fase se prueba la solución implementada.

Documentación: En esta fase se documenta la solución implementada.

Metodologías ágiles

Las metodologías ágiles más utilizadas para el diseño de proyectos de análisis y ciencia de datos son:

Scrum: Scrum es una metodología ágil que se basa en el desarrollo iterativo e incremental. Scrum consta de las siguientes fases:

Sprint: Un sprint es un periodo de tiempo, generalmente de dos semanas, durante el cual se desarrolla un conjunto de funcionalidades de la solución.

Planificación: En la planificación del sprint se identifican las funcionalidades que se desarrollarán durante el sprint.

Desarrollo: En el desarrollo del sprint se implementan las funcionalidades identificadas en la planificación.

Revisiones: En las revisiones se revisa el progreso del sprint y se identifican los riesgos.

Retrospectivas: En las retrospectivas se analizan los resultados del sprint y se identifican las mejoras que se pueden realizar.

Kanban: Kanban es una metodología ágil que se basa en el flujo continuo de trabajo. Kanban consta de los siguientes elementos:

Tarjetas: Las tarjetas representan las tareas o funcionalidades que hay que realizar.

Columnas: Las columnas representan las etapas del flujo de trabajo.

Reglas: Las reglas definen cómo se mueven las tarjetas entre las columnas.

Extreme Programming (XP): XP es una metodología ágil que se centra en la calidad del software. XP consta de los siguientes principios:

Programación en parejas: Los desarrolladores trabajan en parejas para aumentar la comunicación y la colaboración.

Integración continua: El software se integra continuamente para detectar y corregir errores de forma temprana.

Pruebas unitarias: Las pruebas unitarias se utilizan para garantizar la calidad del software.

Revisión de código: El código se revisa regularmente para garantizar su calidad.

Entregas frecuentes: El software se entrega con frecuencia para obtener feedback de los usuarios.

La elección de la metodología adecuada para el diseño de un proyecto de análisis y ciencia de datos depende de una serie de factores, como el tamaño y la complejidad del proyecto, los requisitos del cliente y las preferencias del equipo de desarrollo.

Referencias bibliográficas

Metodologías tradicionales

Modelo en cascada:

Pressman, R. S. (2018). Ingeniería del software: un enfoque práctico (8.ª ed.). McGraw-Hill.

Modelo en espiral:

Boehm, B. W. (2000). The spiral model: a metamodel of software development. IEEE Computer, 23(5), 61-72.

Modelo V:

Pressman, R. S. (2018). Ingeniería del software: un enfoque práctico (8.ª ed.). McGraw-Hill.

Metodologías ágiles

Scrum:

Schwaber, K., & Sutherland, J. (2017). Scrum: The Art of Doing Twice the Work in Half the Time (3.ª ed.). ScrumGuides.

Kanban:

Anderson, D. (2016). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.

Extreme Programming (XP):

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Highsmith, J. (2001). The Agile Manifesto. Agile Alliance.

Además de estas referencias, también se puede consultar la siguiente bibliografía:

Bates, A. (2018). Agile Data Science. O'Reilly Media.

Gartner. (2022). Gartner Market Guide for Data Science and Machine Learning Platforms. Gartner.

McKinsey & Company. (2022). The State of Data Science in the Enterprise. McKinsey & Company.

#### ¿Que son los datos, información y conocimiento? Realiza un ejemplo.

Los datos sin procesar no tienen un significado por si solos.

En la información los datos ya tienen un valor que significativo y legible.

Y el conocimiento se convierte en la información que es comprendida y aplicable.

Ejemplo: En una encuesta sé pregunta que genero de videojuego le gusta más a la gente, se recibe la información que dan los entrevistados que se convierten en datos, luego pasa por un proceso de análisis en donde a los datos obtenidos se empieza a obtener una información sobre cual fue le mas botado y gano los juegos en primera persona, y por último la información se ponen en conocimiento a empresas que se dediquen al desarrollo de videojuegos lo que hace esas empresas comprendan que es lo mas le gusta ala gente para aplicarlo en futuros productos.

##### ¿Como se debe construir un motor de análisis?

La construcción de un motor de análisis es un proceso complejo que requiere la participación de un equipo de expertos en diferentes disciplinas, como la ciencia de datos, la ingeniería de software y el diseño de sistemas.

En general, el proceso de construcción de un motor de análisis se puede dividir en las siguientes etapas:

Definición de los requisitos: En esta etapa, se identifican los requisitos funcionales y no funcionales del motor de análisis. Los requisitos funcionales definen lo que el motor debe hacer, mientras que los requisitos no funcionales definen cómo debe hacerlo.

Diseño: En esta etapa, se diseña la arquitectura del motor de análisis. La arquitectura define la estructura y el funcionamiento del motor.

Implementación: En esta etapa, se implementa el motor de análisis de acuerdo con el diseño.

Pruebas: En esta etapa, se prueba el motor de análisis para garantizar que cumpla con los requisitos.

Implementación: En esta etapa, se implementa el motor de análisis en producción.

Referencias:

* Bates, A. (2018). Agile Data Science. O'Reilly Media.
* Gartner. (2022). Gartner Market Guide for Data Science and Machine Learning Platforms. Gartner.
* McKinsey & Company. (2022). The State of Data Science in the Enterprise. McKinsey & Company.

##### ¿Qué es el mapa VD? Brinda un ejemplo y Explícalo.

El VD sirve para representar los procesos ver oportunidades de mejora y comunicar el proceso a los demás.

Ejemplo: En una empresa se va desarrollar un nuevo producto pero no saben como cuadrar al equipo que va a trabajar en ese nuevo producto así que se toma la decisión de acomodar en un mapa VD, primero se planea el producto que se va hacer, se pasa al diseño y se evalúa que tan factible el diseño se hace la entrega para la producción y por ultimo se hace el proceso de apoyo que sería de promocionarla.

##### ¿Que son los escenarios funcionales y de calidad? Realiza un cuadro comparativo.

Estos escenarios comentan como debe funcionar el sistema y como debe de cumplir el sistema.

|  |  |
| --- | --- |
| Escenarios funcionales. | Escenarios de calidad |
| Como debe funcionar el sistema.  Un usuario inicia sesión en el sistema. | Calidad que el sistema debe cumplir.  El sistema debe procesar 100 inicios de sesión por segundo. |

##### Explica que es la GUI y el material design.

La GUI es una interfaz de usuario que permite al usuario interactuar con el sistema y el material design es un lenguaje de desarrollo visual creado por Google para una experiencia de los usuarios consistente y coherente en todas la plataformas y dispositivos de Google.

##### Explica que es el modelo UI y UX. Realiza un ejemplo.

Estos modelos indican la satisfacción del usuario y la usabilidad que el mismo le da.

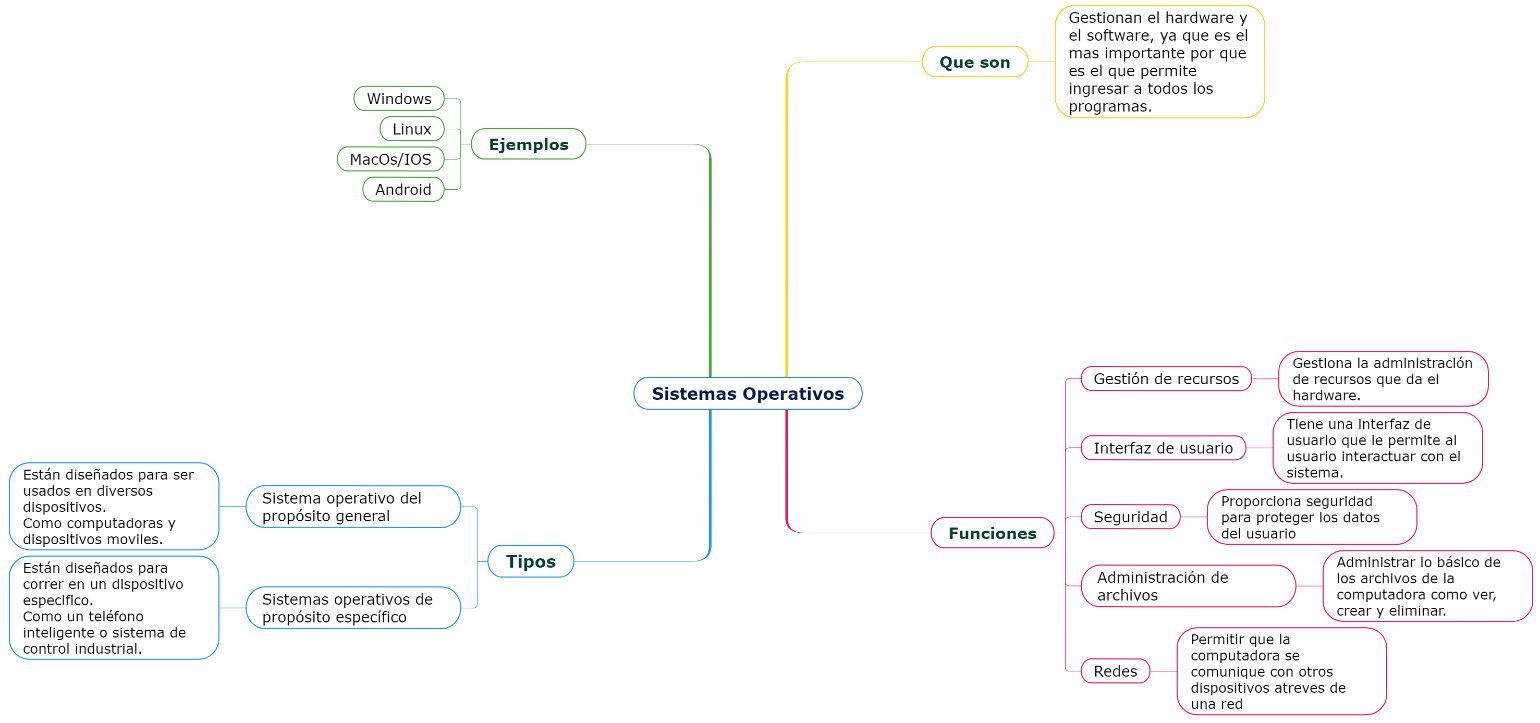
La UI es la interfaz de lo que ve el usuario y si es atractiva y fácil de usar.

La UX Es como el usuario se siente al usar el producto.

##### ¿Por qué son importante los Bocetos manuales y los prototipos digitales? Justifica.

Ambos ayudan a la comprensión del diseño del producto digital que se está desarrollando y que para los que tienen la idea de diseño, pero no saben como hacer les da una idea y entendimiento de esas personas sobre el diseño que se hizo.

##### Explica que es un sistema operativo a través de un mapa mental.



##### ¿Qué es un kernel?

El kernel, o núcleo, es la parte central de un sistema operativo. Es el software que se ejecuta en el nivel más bajo del sistema, y se encarga de gestionar los recursos de hardware y software de la computadora.

Tanenbaum, Andrew S., y Herbert Bos. "Sistemas operativos modernos." (2014).

##### Explica cada una de las capas en que se compone el sistema operativo Android e IOS.

Android

Android se compone de las siguientes capas:

* Núcleo: El núcleo de Android es un kernel de Linux modificado que proporciona las funciones básicas de gestión de recursos, interfaz con el hardware y seguridad.
* Librerías del sistema: Las bibliotecas del sistema proporcionan funciones y servicios a las aplicaciones, como acceso a la memoria, al almacenamiento y a la red.
* Marco de aplicaciones: El marco de aplicaciones proporciona una estructura para que las aplicaciones se ejecuten y comuniquen entre sí. Incluye componentes como las actividades, los servicios, los receptores de eventos y los proveedores de contenido.
* API de aplicaciones: La API de aplicaciones proporciona a los desarrolladores una forma de acceder a las funciones del sistema operativo.
* Aplicaciones: Las aplicaciones son los programas que ejecutan los usuarios. Pueden ser aplicaciones de sistema, que son proporcionadas por Google, o aplicaciones de terceros, que son desarrolladas por otros desarrolladores.

IOS

iOS se compone de las siguientes capas:

* Kernel: El kernel de iOS es un kernel de Darwin que proporciona las funciones básicas de gestión de recursos, interfaz con el hardware y seguridad.
* Bibliotecas del sistema: Las bibliotecas del sistema proporcionan funciones y servicios a las aplicaciones, como acceso a la memoria, al almacenamiento y a la red.
* Marco de aplicaciones: El marco de aplicaciones proporciona una estructura para que las aplicaciones se ejecuten y comuniquen entre sí. Incluye componentes como las vistas, los controladores y los modelos.
* API de aplicaciones: La API de aplicaciones proporciona a los desarrolladores una forma de acceder a las funciones del sistema operativo.
* Aplicaciones: Las aplicaciones son los programas que ejecutan los usuarios. Pueden ser aplicaciones de sistema, que son proporcionadas por Apple, o aplicaciones de terceros, que son desarrolladas por otros desarrolladores.

Bibliografía:

* Android Open-Source Project documentation.
* Apple iOS documentation.
* Tanenbaum, Andrew S., y Herbert Bos. "Sistemas operativos modernos." (2014).

##### Explique las dos principales arquitecturas de kernel.

* Kernel monolítico: El kernel monolítico es el tipo de kernel más común. Es un solo programa grande que contiene todas las funciones del sistema operativo, incluidas la gestión de recursos, la interfaz con el hardware y la seguridad.
* Kernel microkernel: El kernel microkernel es un tipo de kernel más pequeño que separa las funciones del sistema operativo en módulos independientes. Esto hace que el kernel sea más modular y fácil de mantener.

##### ¿Qué es un framework? ¿Cuál es la importancia de estos? Menciona las ventajas y desventajas.

Un framework es un conjunto de herramientas, bibliotecas y convenciones que se utilizan para desarrollar software de una manera más eficiente y consistente. Los frameworks proporcionan una estructura básica para que los desarrolladores puedan centrarse en la lógica empresarial de su aplicación, en lugar de tener que implementar las funcionalidades comunes desde cero.

La importancia de los frameworks radica en que pueden ahorrar a los desarrolladores una gran cantidad de tiempo y esfuerzo. Al proporcionar una estructura básica, los frameworks permiten a los desarrolladores centrarse en la lógica empresarial de su aplicación, en lugar de tener que implementar las funcionalidades comunes desde cero. Esto puede conducir a un desarrollo más rápido y eficiente, y a aplicaciones de mayor calidad.

Algunas de las ventajas de los frameworks incluyen:

* Ahorro de tiempo y esfuerzo: Los frameworks pueden ahorrar a los desarrolladores una gran cantidad de tiempo y esfuerzo, ya que proporcionan una estructura básica para desarrollar aplicaciones.
* Mejor calidad: Los frameworks pueden ayudar a los desarrolladores a crear aplicaciones de mayor calidad, ya que proporcionan una base sólida para construir sobre ella.
* Mayor flexibilidad: Los frameworks pueden ser flexibles y adaptarse a las necesidades específicas de las aplicaciones.
* Colaboración: Los frameworks pueden facilitar la colaboración entre los desarrolladores, ya que proporcionan una base común para trabajar.

Algunas de las desventajas de los frameworks incluyen:

* Cierta rigidez: Los frameworks pueden ser rígidos en cuanto a la forma en que se deben desarrollar las aplicaciones.
* Dependencia: Los frameworks pueden crear una dependencia en el framework, lo que puede dificultar la migración a otro framework o la implementación de nuevas funcionalidades.
* Coste: Los frameworks pueden ser costosos de adquirir y mantener.

(Google,sf)