



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL
FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- Denominación del Programa de Formación: ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN
- Código del Programa de Formación: 228106 V102
- Nombre del Proyecto: DISEÑO Y CONSTRUCCIÓN DE SOLUCIONES INFORMÁTICAS PARA GRANDES VOLÚMENES DE DATOS EN EL SECTOR PRODUCTIVO 1787645
- Fase del Proyecto: DISEÑO, DESARROLLO
- Actividad de Proyecto:
 - DISEÑAR LA ESTRUCTURA TECNOLÓGICA DEL SISTEMA DE INFORMACIÓN
 - PREPARAR EL PROCESO DE IMPLANTACIÓN DEL SISTEMA DE INFORMACIÓN
- Competencia:
 - 220501007 - CONSTRUIR EL SISTEMA QUE CUMPLA CON LOS REQUISITOS DE LA SOLUCIÓN INFORMÁTICA.
 - 220501034 - IMPLANTAR LA SOLUCIÓN QUE CUMPLA CON LOS REQUISITOS PARA SU OPERACIÓN.
- Resultados de Aprendizaje Alcanzar:
 - RAP31 - EJECUTAR Y DOCUMENTAR LAS PRUEBAS DEL SOFTWARE, APLICANDO TÉCNICAS DE ENSAYO-ERROR, DE ACUERDO CON EL PLAN DISEÑADO Y LOS PROCEDIMIENTOS ESTABLECIDOS POR LA EMPRESA
 - RAP54 - ELABORAR EL INFORME ADMINISTRATIVO, SIGUIENDO LOS PROTOCOLOS DE LA ORGANIZACIÓN, BASADO EN LOS PLANES DE INSTALACIÓN, RESPALDO Y MIGRACIÓN DEL SISTEMA DE INFORMACIÓN, FACILITANDO LA OPERATIVIDAD Y MANTENIMIENTO DE LA SOLUCIÓN INFORMÁTICA
- Duración de la Guía: 66 horas.



2. PRESENTACIÓN






Después de realizar la fase de DESARROLLO, es necesario realizar las pruebas para certificar que la aplicación cumple con el diseño que se construyó y verificar su funcionalidad (hace lo que debe).

La aplicabilidad de esta guía permitirá el desarrollo de nuevas habilidades en los aprendices del programa de formación relacionadas con técnicas de migración de datos, respaldo de la información y pruebas de software.

Se invita al aprendiz a tomar todos los temas tratados en esta guía para que sea insumo y conocimiento en el desarrollo de sistemas de información.

GFPI-F-135 V01



	Recomendaciones
	El Aprendiz hace
	El instructor explica
	Evidencia por entregar
	Portafolio de evidencias del aprendiz

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial.



Como todo, se desea que los productos que se adquieran tengan calidad; por esta razón el desarrollo de software no es indiferente a esta necesidad, teniendo en cuenta aspectos como: funcionalidad, fiabilidad y el rendimiento, seguridad, resguardo y portabilidad.

- ¿Cuánto tiempo dura una empresa que fabrica aviones, hace pruebas funcionales antes de entregarlo al cliente?
- Justifique el tiempo y diga ¿por qué es importante realizar las pruebas funcionales antes de entregarlo al cliente?

GFPI-F-135 V01



De su opinión en el foro dispuesto en la plataforma LMS sobre las preguntas planteadas.

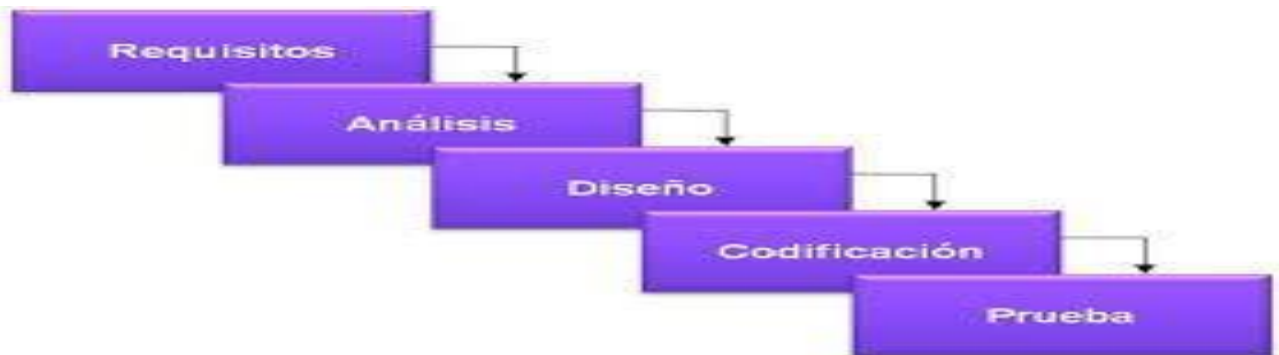
“Todo debe hacerse lo más simple posible. Pero no más sencillo.”
Albert Einstein

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.



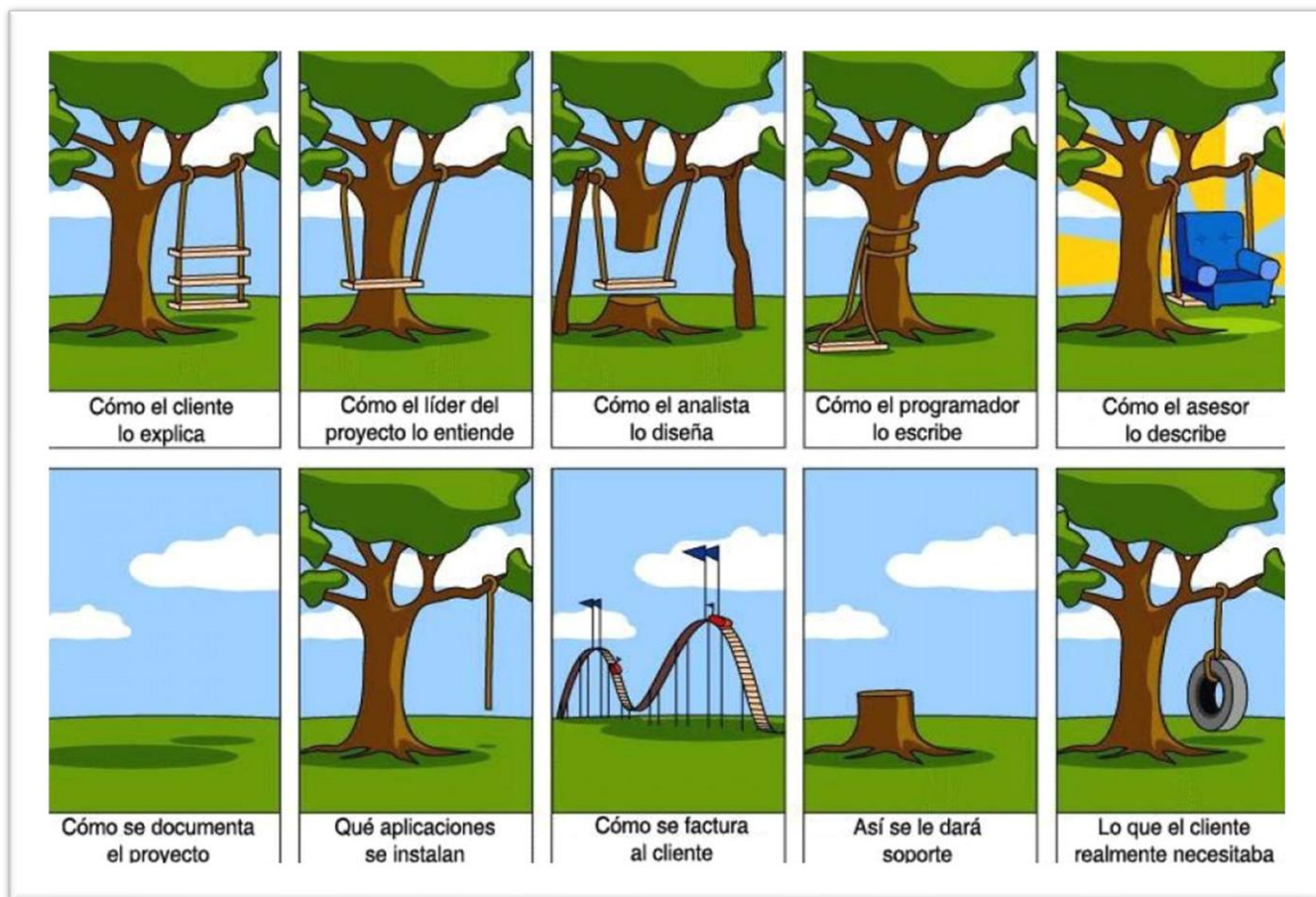
El saber un lenguaje de programación o base de datos no es garantía para que la aplicación haga lo que se debe hacer de acuerdo con los requisitos funcionales y diseños iniciales; por esta razón se deben realizar las pruebas de software que consta de una serie de procesos que permitirán verificar la calidad del software construido para que al final pueda ser implantado o migrada su información. Se trata de que el aprendiz pueda corroborar que su desarrollo y conocimiento del proceso ha sido bien realizado.

Es importante que como aprendiz pueda aplicar conocimientos adquiridos previamente y aplique o evalúe desarrollos propios o de tercero cuando se esté en la etapa de pruebas; en otras palabras, se debe poder entender las fases del ciclo de vida para la construcción de sistemas de información.



Se debe garantizar la seguridad de la información y su resguardo efectivo para cuando ocurra un suceso inesperado o corregir vulnerabilidades y riesgo a éstos.

Nunca se puede resolver un problema en el mismo nivel en el que fue creado.
Albert Einstein



3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).



Examinar el proceso de migración, planes de instalación, respaldo de la información y pruebas de software de acuerdo con normas técnicas.



Cuando se realiza migración de datos o llenado de información básica en los sistemas de información, se debe tener en cuenta lineamientos como se detallan en la [“GUÍA QUARTZ: LIMPIEZA DE DATOS”](#), el cual permite analizar la calidad del dato de la fuente de información.



Se recomienda revisar [“PAUTAS PARA UNA CORRECTA MIGRACIÓN DE BASES DE DATOS”](#) y [“TÉCNICAS DE MIGRACIÓN DE DATOS”](#), que permitirá conocer las diferentes técnicas de migración de datos. Se recomienda ver el video [¿Cómo desarrollar una migración de datos exitosa?](#).



Realizar un documento Word sobre lo visto en el video, súbalo al LMS y sustentarlo en plenaria con los compañeros de la ficha.



No olvide guardar la evidencia en el portafolio del aprendiz.



Observar el video [Migración de datos](#) y desarrollar un mapa conceptual acerca de éste que le permitirá responder el cuestionario de preguntas en el LMS.



Realizar la lectura “[REFLEXIONES SOBRE INGENIERÍA DE REQUISITOS Y PRUEBAS DE SOFTWARE](#)”. Parte 2 sobre “[Verificación y Validación](#)”, pagina 65, dispuesta en la biblioteca del SENA; realizar una presentación

que plasme la lectura realizada.



Para las pruebas de software se revisan las normas internacionales [IEEE-829](#), [IEC 29119](#) y los lineamientos [ISTQB](#).

Lo anterior, se orienta hacia la “[INTRODUCCIÓN A LAS PRUEBAS DE SOFTWARE](#)”, “[TESTING DE SOFTWARE](#)” y “[PRUEBAS](#)”.



Durante la actividad de aprendizaje se utilizan generadores de datos para realizar pruebas, se recomiendan los siguientes sitios web:

- [Mockaroo](#)
- [Generatedata](#)
- [databene benerator](#)

“Una vez que aceptamos nuestros límites, podemos ir más allá de ellos.”
Albert Einstein



Descargar [Postgres](#) para realizar los talleres recomendados para esto leer el manual de [Postgresql](#)



Realizar los talleres: “[MIGRACIÓN DESDE EXCEL POR DML](#)” y “MIGRACIÓN POR ODBC”, para resolver el caso de estudio “CASO DE ESTUDIO COVID19”

**“Debo estar dispuesto a renunciar a lo que soy
con el fin de convertirse en lo que seré.”
Albert Einstein**

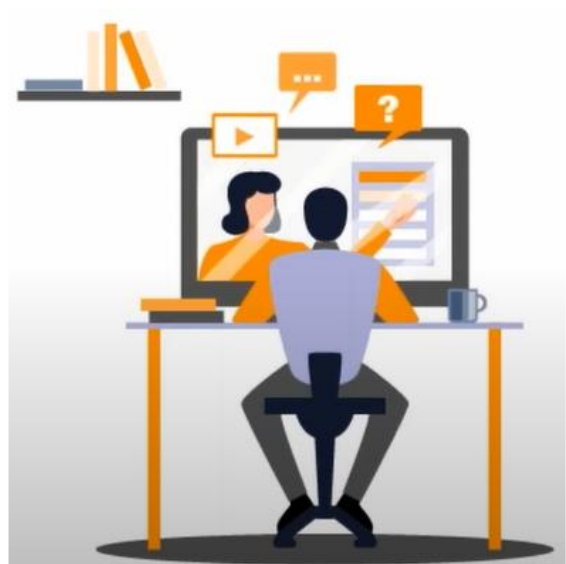
3.3.1 Utilizar migración de datos.

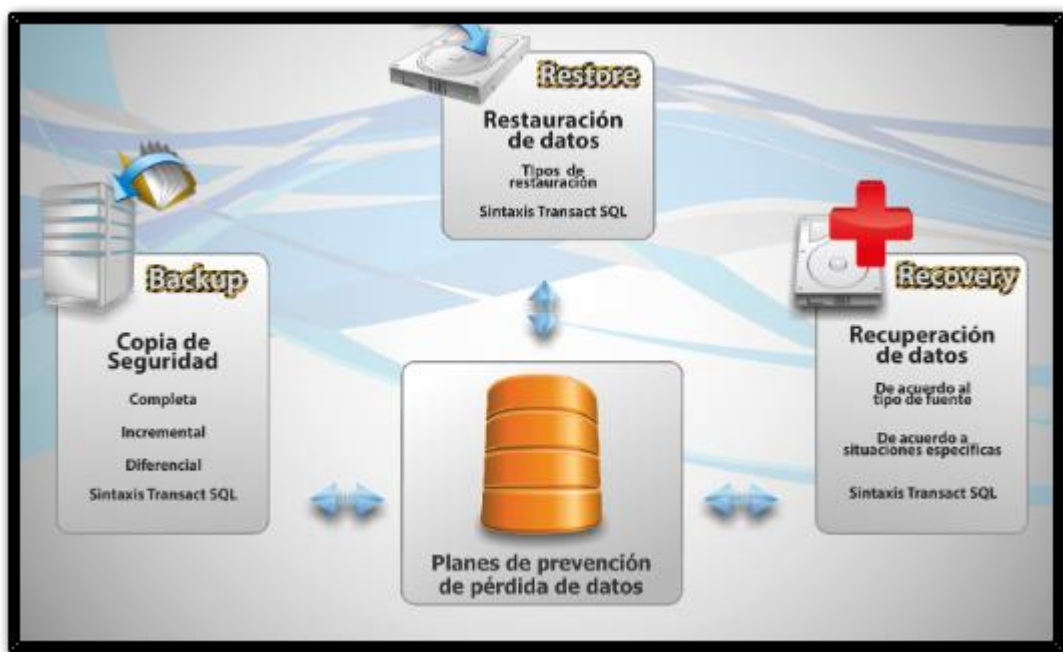


La técnicas [ETCL](#) (Extraction – Transformation- Clear- Load), Extracción – Transformación-Limpieza y Carga, permite realizar migración de datos.

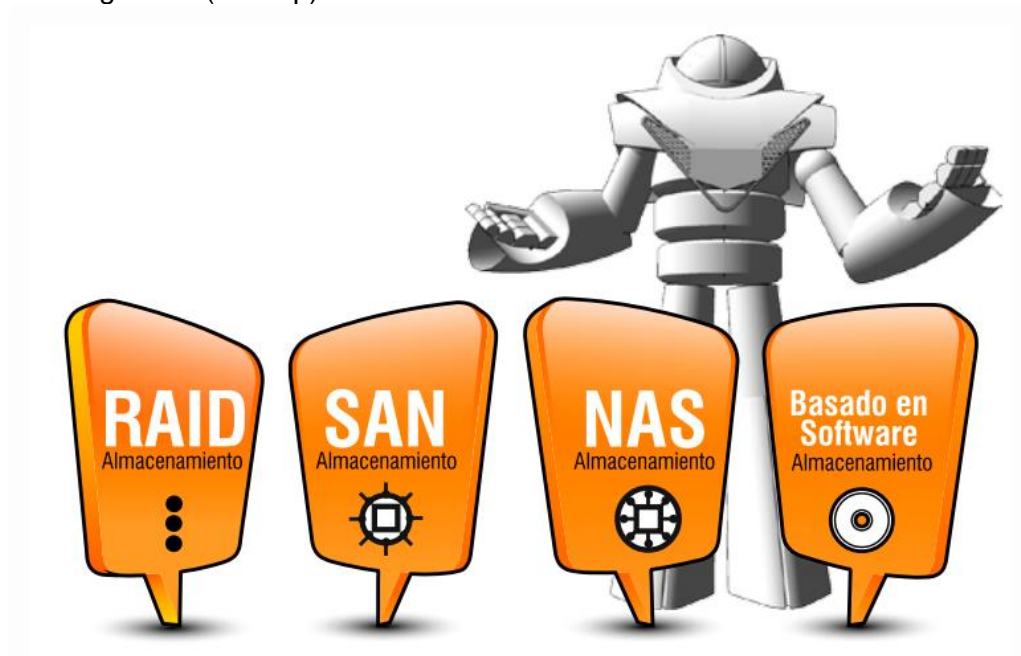


Ver el video “[Estrategia de migración de datos entre sistemas informatizados en sectores regulados](#)”





Realizar la lectura “[COPIAS DE SEGURIDAD](#)”, para reconocer los diferentes tipos de copias de seguridad (backup).



Realizar la lectura “SISTEMA DE ALMACENAMIENTO”, para reconocer los diferentes tipos de almacenamiento “[RAID](#)”, “[SAN](#)”, “[NAS](#)”.



Política de seguridad y resguardo de la información para la continuidad del negocio.



Realizar la lectura “[PLAN DE RECUPERACIÓN DE DESASTRES](#)”, entender y

desarrollar planes de recuperación de desastres (DRP), para si poder realizar:

- Matriz de información a resguardar, cantidad e importancia en caso de pérdida.
- Periodicidad y tipos de resguardo a utilizar, como también medios de almacenamiento
- Plan de recuperación en el momento de caída o pérdida de datos en el sistema.

Ver el video “[PLAN DE RECUPERACIÓN DE DESASTRES \(DRP EN AWS\)](#)” y “[WEBINAR: RECUPERACIÓN DE](#)



[DE DESASTRES INFORMÁTICOS \(DRP\)](#)” y “[WEBINAR: RECUPERACIÓN DE DESASTRES \(DRP EN AWS\)](#)”.

**“La debilidad de actitud se vuelve debilidad de carácter.”
Albert Einstein**

3.4 Actividades de transferencia del conocimiento

Los materiales de las practicas se encuentran en <https://github.com/fegasu/TESTING>

3.4.1 Responder Cuestionario de preguntas acerca de migración de datos de acuerdo con estándares internacionales.



Se dispone de un cuestionario de términos de migración de datos ubicado en el LMS.

**“Somos arquitectos de nuestro propio destino.”
Albert Einstein**

3.4.2 Responder Cuestionario pruebas de software de acuerdo con los estándares ITSQB



Se dispone de un cuestionario de términos de pruebas de software ubicado en el LMS.

**“La creatividad es la inteligencia divirtiéndose.”
Albert Einstein**

3.4.3 Desarrollar el Caso de estudio acerca del COVID19 con estándares y requerimientos de la organización.

GFPI-F-135 V01



El grupo de investigación del SENA llamado “Desafíos” ha solicitado que se cargue en una base de datos los datos abiertos del ministerio de salud de Colombia sobre la pandemia [COVID19](#) que se encuentra en los archivos adjunto, después de descargar y descomprimir se debe verificar la calidad de los datos, lo mismo que datos perdidos, problemas ortográficos, como los datos entregados están en primera forma normal (1FNN), debemos llevar hasta la tercera forma normal (3FN) y crear las respectivas tablas que se generaron en la fase de [NORMALIZACIÓN](#).



Para esto se hace necesario realizar lo siguiente:

- Construir un diseño de base de datos del [TIPO ESTRELLA](#) que permita migrar los datos de la fuente de datos del [COVID19](#), mediante una tabla de hecho y sus dimensiones.
- Realizar el modelo físico de acuerdo con el diseño de base datos de la fuente de datos.
- Hacer el plan de migración para la migración de los datos desde la fuente información.
- Revisar la calidad de la migración mediante un plan de pruebas.
- Construir los casos de prueba para aplicar en los casos de prueba.



Para desarrollar la actividad del caso de estudio propuesto se debe realizar el taller “[CASO DE ESTUDIO COVID19](#)” acompañado con el instructor que le guíara con los conocimientos necesarios para lograr construir la evidencia.



Esta evidencia se debe elaborar en grupo de tres aprendices en las fechas acordadas en el plan de trabajo concertado (PTC) y las instrucciones dadas por el instructor.

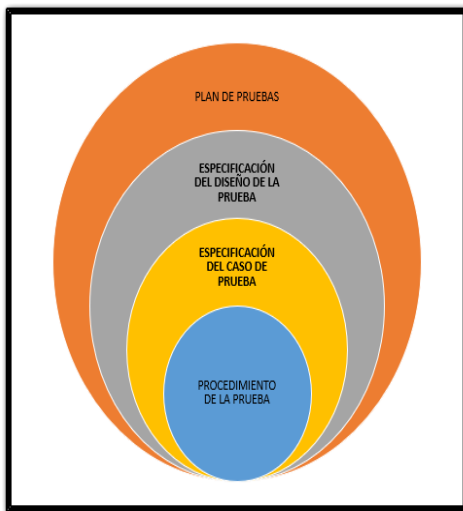


No olvide guardar la evidencia en el portafolio del aprendiz.

“Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”
Albert Einstein



3.4.4 Desarrollar Caso de estudio ALQUILAPC de acuerdo con estándares y requerimientos de la organización.



Descargar el programa [TestLink](#) para realizar las pruebas de software, se debe tener instalado [XAMPP versión 5.6.14](#), se recomienda leer el manual de [TestLink](#).

Descripción del caso de estudio



Una vez se ha leído el material adjunto y siguiendo las indicaciones del “TALLER ALQUILAPC”, se puede desarrollar el caso de estudio.



El caso de estudio está compuesto de tres partes: La primera parte consiste en una prueba de Caja negra donde se entregara una aplicación que se ejecuta y donde se darán unas reglas de negocio que se deben hacer pruebas para confirmar si está de acuerdo a ellas; se deben realizar casos de

prueba necesario para su evaluación y reportar donde se encuentran las fallas; la segunda parte es una prueba de caja blanca, donde se contara con el fuente en un lenguaje de programación, donde se debe ajustar para que el programa funcione de forma exacta a las reglas de negocio planteada por la empresa ALQUILAPC y la tercera y última parte es la prueba de regresión, donde la empresa ALQUILAPC cambia las reglas de negocio y de deben realizar los ajuste requerido en el programa y validar que los resultados estén de acuerdo a los casos de pruebas planteados. No olvide [VALIDAR LOS DATOS](#).



Llenar el documento “[PLAN DE PRUEBAS.docx](#)” por cada prueba realizada, cuando se termine de desarrollar el caso de estudio.



Este documento debe ser entregado en la fecha acordada en el plan de trabajo concertado (PTC) y se debe llamar “**CasoDeEstudioAlquilaPc.zip**”



No olvide guardar la evidencia en el portafolio del aprendiz.

**“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.”
Albert Einstein**



Realizar el taller de [pruebas unitarias automatizadas con JUNIT](#).



No olvide guardar una copia en el portafolio de aprendizaje con el registro gráfico de la práctica.

**“En medio de la dificultad reside la oportunidad.”
Albert Einstein**



4. ACTIVIDADES DE EVALUACIÓN

Materiales de formación [AQUÍ](#)

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento:		
3.4.1 Contesta Cuestionario de preguntas acerca de migración de datos	ELABORA EL INFORME ADMINISTRATIVO DE LA SOLUCIÓN INFORMÁTICA IMPLANTADA, DE ACUERDO CON LOS PLANES DE INSTALACIÓN, RESPALDO Y MIGRACIÓN DEL SISTEMA ESTIPULADOS POR LA ORGANIZACIÓN, UTILIZANDO NORMAS Y PROTOCOLOS	IEV1. Cuestionario
3.4.2 Contesta Cuestionario pruebas de software de con acuerdo a los estándares ITSQB		IEV2. Cuestionario
Evidencias de Desempeño.		
3.4.3 Desarrolla el Caso de estudio de migración de datos.	REALIZA Y DOCUMENTA LAS PRUEBAS DEL SOFTWARE SEGÚN EL PLAN ESTABLECIDO, PARA DEFINIR ACCIONES CORRECTIVAS A SEGUIR Y ASEGURAR EL BUEN FUNCIONAMIENTO DEL SISTEMA DE INFORMACIÓN	IEV3. Lista de verificación
Evidencias de Producto:		
3.4.4 Desarrolla Plan de pruebas de software a un caso de estudio.	REALIZA PLAN DE PRUEBA UTILIZANDO HERRAMIENTA DE PRUEBAS DE SOFTWARE	IEV4. Lista de verificación

5. GLOSARIO DE TÉRMINOS

- **Anomalía:** Cualquier condición que se desvíe de las expectativas basadas en las especificaciones de requisitos, documentos de diseño, documentos de usuario, estándares, etc., o de la percepción o experiencia de alguien. Las anomalías pueden ser encontradas durante, aunque no se limitan sólo a, revisiones, proceso de pruebas, análisis, compilación, o uso de productos de software o documentación aplicable. [IEEE 1044] Véase también bug, defecto, desviación, error, falta, fallo, incidente, problema.
- **Criterios de aceptación:** Los criterios de salida que un componente o sistema debe satisfacer para ser aceptado por un usuario, cliente u otra entidad autorizada. [IEEE 610]

GFPI-F-135 V01



- Caso de prueba: Conjunto de condiciones, datos o variables que servirán para determinar si los requisitos del sistema se cumplen de manera parcial, completa, o no se cumplen.
- Conjunto de pruebas base: Conjunto de casos de prueba derivados de la estructura interna de un componente o especificación para asegurar que será alcanzado el 100% de un criterio de cobertura especificado.
- Error: Discrepancia entre el valor calculado y el valor teórico o esperado, con responsabilidad del desarrollador.
- Exactitud: Capacidad de un producto software de proporcionar el resultado o efecto correcto o acordado con el grado de precisión requerido. [ISO9126]. Véase también pruebas de funcionalidad.
- Defecto software: Desviación en el valor esperado por una cierta característica. Defecto de calidad.
- Fallo: Consecuencia de un error o un defecto software.
- Línea base: Especificación o producto software que ha sido formalmente revisada o para la cual se ha llegado a un acuerdo, que en adelante sirve como base para un desarrollo posterior y que puede ser modificado sólo a través de un proceso formal de control de cambios. [Según IEEE 610]
- pruebas ad hoc: Pruebas llevadas a cabo de manera informal; no se realiza una preparación formal de la prueba, no se utilizan técnicas de diseño reconocidas, no existen expectativas para con los resultados y la arbitrariedad guía la actividad de ejecución.
- Prueba: Proceso mediante el cual se ejecuta de manera sistemática un conjunto de actividades (métodos y técnicas) para encontrar errores.
- Pruebas ágiles: Práctica de pruebas para un proyecto que utiliza metodologías ágiles, tales como programación extrema (XP), tratando al desarrollo como el cliente del proceso de pruebas y enfatizando el paradigma de diseño de primero-probar. Véase también desarrollo dirigido por las pruebas.
- Pruebas de aceptación: Pruebas formales con respecto a las necesidades de usuario, requisitos y procesos de negocio dirigidas a determinar si el sistema satisface o no los criterios de aceptación y a habilitar al usuario, cliente u otra entidad autorizada a determinar si acepta o no el sistema. [Según IEEE 610]
- Pruebas de accesibilidad: Pruebas para determinar la facilidad con la que un usuario con discapacidades puede usar un componente o sistema. [Gerrard]
- Pruebas beta: Pruebas operacionales realizadas por usuarios/clientes potenciales y/o existentes, en un sitio externo no relacionado de ninguna manera con los desarrolladores, para determinar si un componente o sistema satisface o no las necesidades del usuario/cliente y se ajusta a los procesos de negocio. Con frecuencia las pruebas beta se emplean como una forma de prueba de aceptación externa para software de distribución masiva con el objetivo de obtener la respuesta del mercado.
- Pruebas tipo big-bang: Tipo de prueba de integración en el que los elementos software, elementos hardware ó ambos son combinados de forma simultánea en un componente o un sistema global en lugar de hacerlo por fases. [Según IEEE 610] Véase también pruebas de integración.
- Técnica de diseño de prueba de caja negra: Procedimiento para obtener y/o seleccionar casos de prueba basados en el análisis de la especificación, tanto funcional como no funcional de un componente o sistema sin referencia a su estructura interna.
- Pruebas de caja negra: Pruebas tanto funcionales como no funcionales, sin referencia a la estructura interna del componente o sistema.



- Pruebas ascendentes: Enfoque incremental para pruebas de integración donde los componentes de más bajo nivel son probados en primer lugar, posteriormente son utilizados para facilitar las pruebas de componentes de un nivel superior. Este proceso se repite hasta que es probado el componente en el extremo superior de la jerarquía. Véase también pruebas de integración.
- Overflow: Significa desbordamiento en el buffer, cuando la cantidad de datos supera la capacidad preasignada. Es un fallo de programación.
- Rastro de auditoría: Camino por el cual se puede recuperar la entrada original a un proceso (por ejemplo datos) a través del mismo (proceso), tomando como punto de partida la salida del proceso. Esto facilita el análisis de defectos y permite llevar a cabo la auditoría de un proceso. [Según TMap]
- Underflow: Significa subdesbordamiento del buffer, cuando se carga datos a una velocidad inferior a la de procesamiento, provocando bloqueos.
- Valor límite: Valor de entrada o de salida que se encuentra en la frontera de una partición de equivalencia o a la mínima distancia incremental a cualquier lado de la frontera, por ejemplo el valor mínimo o máximo de un rango.

6. REFERENTES BIBLIOGRÁFICOS

- BOLAÑOS, D., SIERRA, A., & ALARCÓN, M. (2008). Pruebas de Software y JUnit. Madrid: Pearson Prentice Hall.
- CATALDI, Z. (2000). Metodología de diseño, desarrollo y evaluación de software educativo. Tesis de Magíster en Informática. Argentina: Facultad de Informática. Universidad Nacional de la Plata (UNLP).
- PRESSMAN, R. (2006). Ingeniería del Software: Un enfoque práctico. Sexta edición. McGrawHill.
- Bustamante Martínez, A., Galvis Lista, E. A., & Gómez Flórez, L. C. (2013). Técnicas de modelado de procesos de ETL: una revisión de alternativas y su aplicación en un proyecto de desarrollo de una solución de BI. *Scientia et technica*, 18(1).
- El-Sappagh, S. H. A., Hendawi, A. M. A., & El Bastawissy, A. H. (2011). A proposed model for data warehouse ETL processes. *Journal of King Saud University-Computer and Information Sciences*, 23(2), 91-104.
- Natsys (2017). Testing: Convertite en un experto en QA.
- Bath, G. and Veenendaal, E. (2013). Improving the test process : Implementing improvement and change - A study guide for the ISTQB expert. 1st ed. Rocky Nook.
- Black, R. and Rueda Sandoval, G. (2011). Fundamentos de prueba de software. 1st ed. Texas: RBCS.
- Black, R. (2015). Advanced Software Testing - Vol. 1: Guide to the ISTQB Advanced Certification as an Advanced Test Analyst. 2nd ed. Rocky Nook.



- Test Maturity Model integration (TMMi). (2015). [ebook] Erik van Veenendaal. Disponible en: <https://www.tmmi.org/wp-content/uploads/2016/09/TMMiFramework-R1.0-Spanish.pdf>.
- Román, I. R., & Cosín, J. D. (2007). Técnicas Cuantitativas para la Gestión en la Ingeniería del Software: Netbiblo.
- Myers, G. J., Sandler, C., Badgett, T., & Thomas, T. M. (2004). The Art of Software Testing: Wiley.
- Avella Ibañez, C., Gomez Estupiñán, J. and Caro Pineda, S. (2011). Aplicación de inspecciones y pruebas de software. 1st ed. Tunja: Universidad de Boyacá.
- Black, R. (2014). Advanced software testing. : Volume 2, Guide to the ISTQB advanced certification as an advanced test manager. 2nd ed. Rocky Nook.
- Black, R. (2016). Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional: Wiley.
- Mitchell, J. and Black, R. (2015). Advanced Software Testing Vol. 3: Guide to the ISTQB Advanced Certification as an Advanced Technical Test Analyst. 2nd ed. Rocky Nook.
- Linz, T. (2014). Testing in Scrum: a guide for software quality assurance in the agile world. 1st ed. Rocky Nook.
- Montoya, E. S. (2013). Prueba funcional del software: un proceso de verificación constante: Instituto Tecnológico Metropolitano.
- Software Quality Assurance. (2019). Retrieved 3-10-2019, 2019, from <https://greensqa.com/>
- IEEE 829:2008 - Standard for Software and System Test Documentation. (2008). [ebook] New York: The Institute of Electrical and Electronics Engineers, Inc. Disponible en: https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=6054.
- DESAI, S., & SRIVASTAVA, A. (2016). SOFTWARE TESTING : A Practical Approach: Phi Learning.



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	JOSE FERNANDO GALINDO SUAREZ	INSTRUCTOR	CGMLTI	20/06/2020

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

