**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| PROGRAMA DE FORMACIÓN | Programación de *software* |
| --- | --- |

| COMPETENCIA | 220501096 - Desarrollar la solución de *software* de acuerdo con el diseño y metodologías de desarrollo. | RESULTADOS DE APRENDIZAJE | 220501096-03 - Codificar el *software* empleando el lenguaje de programación seleccionado. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | 15 |
| --- | --- |
| NOMBRE DEL COMPONENTE FORMATIVO | Pruebas unitarias |
| BREVE DESCRIPCIÓN | Este componente formativo aborda aspectos clave de las pruebas unitarias en la programación de herramientas de *software*. Con su estudio responsable el aprendiz se afianzará en la conceptualización básica y en la configuración de herramientas de pruebas, tales como plantillas, modelos, plan de pruebas, informe de pruebas y construcción de *test* unitarios con base en la normativa. |
| PALABRAS CLAVE | Metodología, plantillas, pruebas, *stakeholders*, técnicas. |

| ÁREA OCUPACIONAL | 2 - Ciencias Naturales, aplicadas y relacionadas |
| --- | --- |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS**

**Introducción**

* 1. **Conceptos generales**
  2. **Configuración de herramientas**

2.1. Plantillas

2.2. Plantilla del plan de pruebas de *software*

2.3. Modelo de informe de ejecución de pruebas

* 1. **Tipos de pruebas de *software***
  2. **Construcción de *test* unitarios**

**Introducción**

Aquí comienza el estudio del componente formativo **“Pruebas unitarias”**, un recorrido temático pensado para su avance. Comience por visualizar el video que se muestra enseguida.

¡**Adelante**!



1. **DESARROLLO DE CONTENIDOS** 
   * + 1. **Conceptos generales**

La caracterización de un caso de prueba se refleja cuando tiene una entrada conocida y una salida esperada, lo cual se debe haber formulado mediante un escrito antes de que se ejecute la prueba.



La entrada conocida se debe determinar con unas precondiciones establecidas en el documento, el cual se desarrollará probando una poscondición de la salida esperada.

En lo referente a la caracterización de casos de prueba tenga en cuenta los siguientes aspectos:



**Técnicas de diseño de casos de prueba**

Las fases de prueba son muy importantes en el desarrollo del *software*. Las pruebas ayudan a **determinar la calidad del producto**, también lo modifican y actualizan para la facilidad y uso del usuario final.



Los casos de prueba son bloques construidos, fundamentales para combinarlos y lograr conformar la fase de prueba.

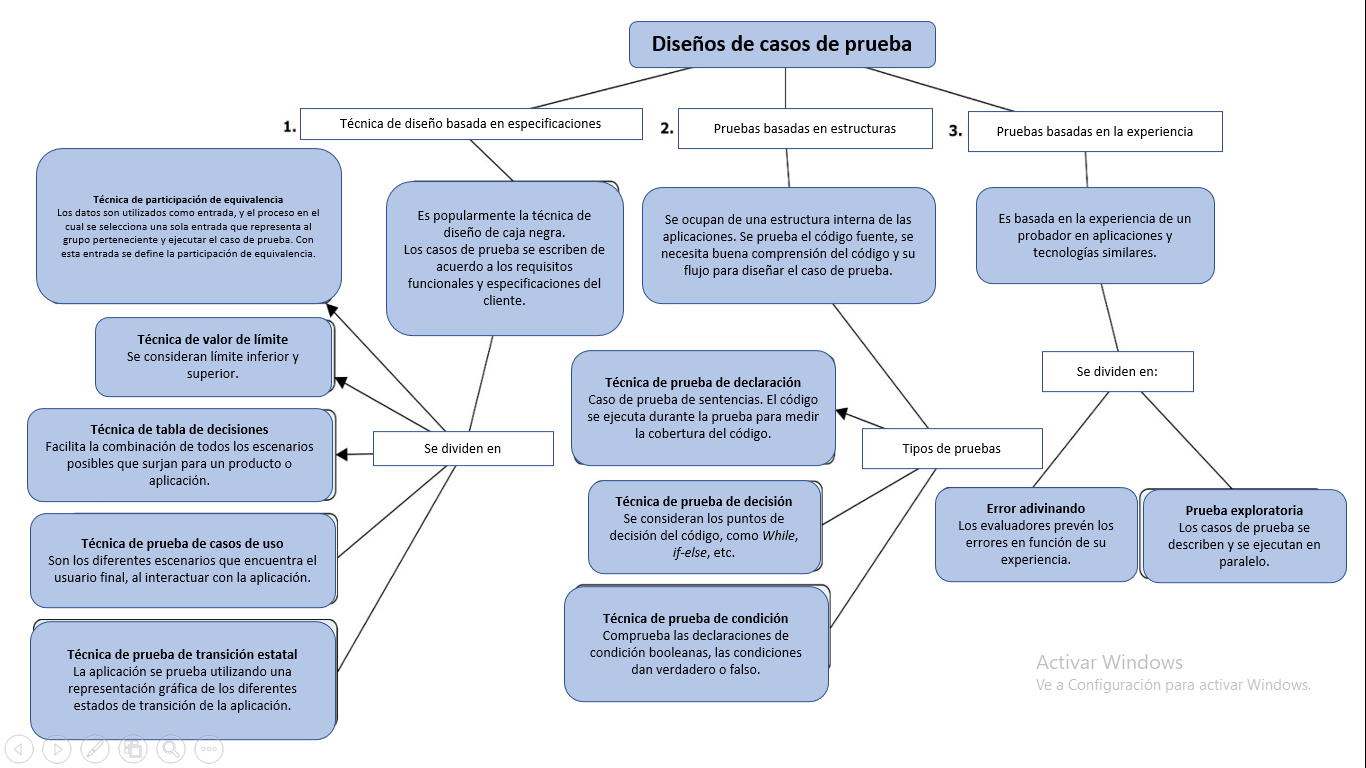
Algunos puntos que debe tener presente sobre las técnicas de diseño de casos de prueba son:



Analice la figura que se muestra enseguida y afiance sus conocimientos sobre el diseño de casos de prueba. Tenga en cuenta los elementos más destacados que allí se mencionan y lleve registro de ello en su libreta personal de apuntes.

**Figura 1**

*Diseño de los casos de pruebas*



Otros tipos de técnicas de pruebas son:

* Prueba de humo vs. Prueba de cordura.
* Prueba de selenio.
* Prueba de caja blanca.
* Prueba de caja gris.
  + - 1. **Configuración de herramientas**

Para realizar cualquier aplicación o producto de *software* el equipo de desarrollo debe guiarse con una serie de pasos que se ajustan al ciclo de vida del *software* SDLC.



Por lo tanto, las pruebas hacen parte de una de las fases, donde se tiene que garantizar la calidad del producto o aplicación y se logra comparar ese funcionamiento con diferentes tipos de parámetros, basados en el diseño y la ejecución de las pruebas de *software*.



**2.1. Plantillas**

Se deben utilizar los elementos útiles al momento de ejecutar los proyectos de desarrollo de *software*. Los elementos a utilizar son plantillas de gerencia de proyectos como:

* Actas de proyecto.
* Definiciones de alcance.
* Cronogramas.
* Tablas de Gantt.
* Planes de gestión de riesgos.
* Planes de comunicaciones, entre otras.

Las plantillas de gerencia de proyectos son importantes porque ayudan a ahorrar tiempo, se evita cometer errores y se selecciona un tipo de metodología adecuada.

Las características de las plantillas para la gerencia de proyectos son:



La Oficina de Gestión de Proyectos PMO desarrolla la metodología que se necesita para gestionar, de muy buena manera, los proyectos empresariales. De igual modo comparte buenas prácticas, lineamientos y metodologías, promoviendo una cultura de gestión de proyectos en todas las empresas. El ideal es capacitar y entrenar a los líderes de los proyectos junto a sus equipos.



Por su parte y como herramientas clave del desarrollo de *software* para la gerencia de proyectos están también las plantillas de *curriculum vitae*, estas son:

* Modelo de *curriculum vitae* para el gerente de proyecto (por logros).
* Modelo de *curriculum vitae* para el gerente de proyecto (por responsabilidades).
* Modelo de *curriculum vitae* para el analista programador.
* Modelo de *curriculum vitae* para el analista de QA.

En el mismo sentido y con el mismo nivel de utilidad están las plantillas de gerencia de proyectos que se enuncian a continuación:

**Tabla 1**

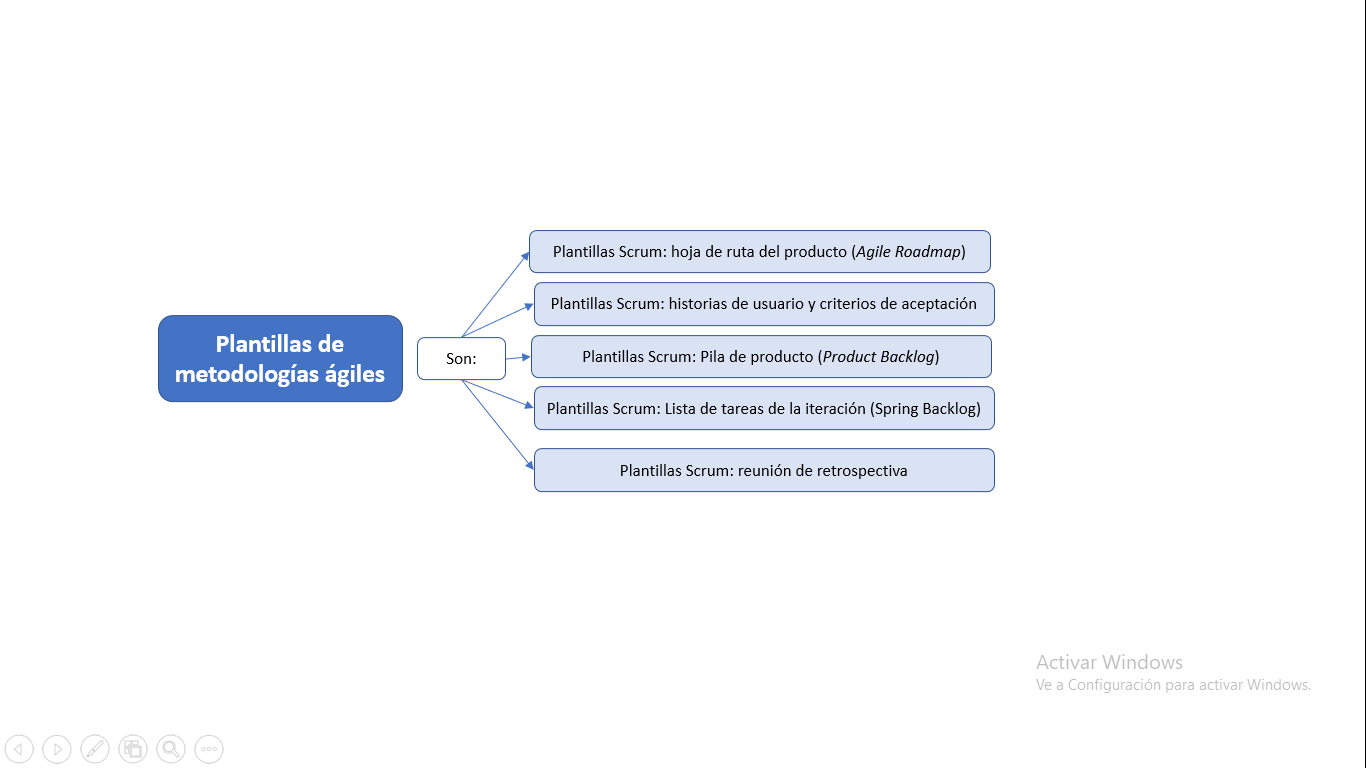
*Plantillas de gerencia de proyectos*

| **Servicios** | Modelo de cómo hacer una propuesta de servicios |
| --- | --- |
| **Factibilidad** | Modelo de estudio de factibilidad de un proyecto |
| **Operación** | Modelo de plan de trabajo de un proyecto |
| **Constitución** | Plantilla de acta de constitución de proyecto |
| **Acta** | Plantilla de acta de proyecto resumida |
| **Dirección** | Modelo de plan de dirección de proyecto |
| **Cierre** | Plantilla de acta de cierre de proyecto |
| **Negocio** | Plantilla de caso de negocio para un proyecto |
| **Distribución operativa** | Plantilla de estructura de desglose de trabajo (EDT) |
| **Memoria** | Plantilla para documentar lecciones aprendidas |
| **Trazabilidad** | Plantilla de matriz de trazabilidad de requisitos |
| **Responsabilidades** | Plantilla de matriz RAM de asignación de responsabilidades |
| **Roles** | Plantilla de matriz de RACI de asignación de roles y responsabilidades |
| **Cambios** | Plantilla de solicitud de cambios en proyectos |
| ***Stakeholders*** | Plantilla del registro de los interesados (*Stakeholders*) |
| **Requerimientos** | Plantilla de requerimientos de comunicaciones del proyecto |
| **Partes interesadas** | Plantilla de plan de gestión de los interesados |
| **Riesgos** | Plantilla de plan de gestión de riesgos en proyectos |
| **Presupuesto** | Plantilla de presupuesto de un proyecto |
| **Seguimiento** | Plantilla para el registro y el seguimiento de los riesgos en proyectos |
| **Reportes** | Plantilla de reporte de avances del proyecto |

Para la aplicación de metodologías ágiles existen también algunas plantillas de mucha utilidad como:

**Figura 2**

*Metodologías ágiles, plantillas*



**2.2. Plantilla del plan de pruebas de *software***

Para el desarrollo de *software*, en la fase de pruebas, se puede caracterizar de manera crítica para que se logre asegurar el producto y para que sea enviado al equipo de producción con la calidad esperada por el cliente o el usuario final.



De igual forma se hace necesario e indispensable contar con un muy buen plan de pruebas de *software* para lograr la especificación de las funciones que se prueban, de la manera como serán ejecutadas las pruebas, asignar a los responsables y el cronograma para lograr las entregas finales.





**Secciones de la plantilla del plan de pruebas**

Hacen referencia a los pasos por seguir, en su orden, para la consecución del desarrollo óptimo de las pruebas y verificación de la calidad del producto.

Analice el siguiente esquema y profundice en la comprensión de las secciones de plantilla del plan de pruebas:



**Plantilla de casos de prueba**

El principal producto que se elabora durante la fase de análisis de pruebas en el proyecto de *software* es la especificación del diseño de casos de prueba y los respectivos datos de entrada.



Estos datos se utilizan para evaluar la calidad del producto y ayuda a determinar si cumple con los requisitos solicitados por el cliente.

Por su parte, la plantilla de casos de prueba presenta en su orden el contenido que se menciona a continuación:

* Id.
* Caso de prueba.
* Descripción.
* Fecha.
* Área funcional / subproceso.
* Funcionalidad o característica.
* Datos y acciones de entrada.
* Datos de salida.
* Requerimientos de los ambientes de prueba.

**2.3. Modelo de informe de ejecución de pruebas**

Al momento de ejecutar los proyectos de desarrollo de *software*, en la fase de pruebas *Software Testing* suele volverse crítica y los *stakeholders* (interesados) buscan información inmediata sobre el estado de la calidad del producto que se está ejecutando.



Para este tipo de evento se suele manejar un informe del avance de las pruebas, desarrollo y funciones, el cual representa la situación actual de las pruebas, los errores, los eventos críticos y los casos por ejecutar.



Los siguientes son los datos de la plantilla del plan de pruebas de *software*:

* Código asociado al proyecto.
* Nombre del proyecto.
* Fecha de comienzo planificada.
* Fecha de finalización planificada.
* Casos de prueba.
* Casos planificados.
* Casos exitosos.
* Avance planificado.
* Avance real.
* Desviación.
* Días de desviación.
* Fecha final del pronóstico.
* Casos con incidencia.

En relación con la situación actual de los casos de prueba tenga presente estos elementos:



En relación con la situación actual de defectos considere:

| **Reportados** | Se reportan la totalidad de defectos o incidencias a la fecha del reporte. |
| --- | --- |
| **En análisis** | Se reportan la cantidad de defectos que se encuentran en análisis, aún no son aceptados. |
| **Descartados** | Son los defectos que se descartan porque no aplican en la prueba por el *tester*. |
| **En proceso** | Se refiere a la cantidad de defectos que se analizaron y están nuevamente en desarrollo. |
| **Corregidos** | Son la cantidad de defectos que ya se corrigieron, por lo tanto, ya se puede cerrar el caso. |

* + - 1. **Tipos de pruebas de *software***

Es importante diferenciar los tipos de pruebas de *software* entre manuales y automatizadas. Las manuales se realizan en persona, se hace clic a través de la aplicación, interactuando con el *software* y las herramientas adecuadas para la prueba.



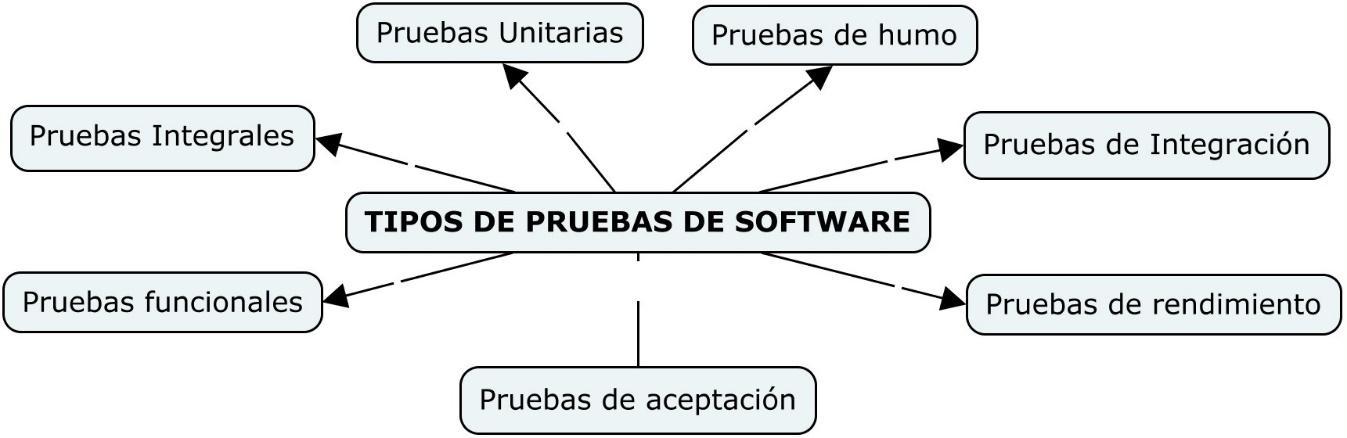
En ocasiones suelen ser muy costosas, puesto que se requiere la configuración del entorno y su ejecución, y suelen estar propensas a errores, ya sea por error humano al omitir alguna funcionalidad o al añadir pasos erróneos de la prueba al ejecutarlo con la aplicación.



En la siguiente figura se muestran los tipos de pruebas de *software*; lleve registro de estos en su libreta personal de apuntes:

**Figura 3**

*Tipos de pruebas de software*



* + - 1. **Construcción de *test* unitarios**

Para lograr determinar la calidad y la funcionalidad de la aplicación se deben diseñar uno o varios casos de prueba donde se integren todas las características posibles, trabajando conjuntamente con el equipo de desarrollo, para así lograr hacer las entregas a tiempo al usuario, velando por su continua funcionalidad y prestación del servicio de *software*.



**Pruebas unitarias de *software***

Este tipo de pruebas son también conocidas como *unit testing* o *test* unitarios. Se describen como el mecanismo de la comprobación del funcionamiento de las unidades de menor tamaño de una aplicación.



Se trata, en definitiva, de trozos de código diseñado para la comprobación de que el código del programa funciona como se espera, según los requerimientos y la funcionalidad del sistema solicitado por el cliente.



En cuanto a las pruebas unitarias tenga en cuenta aspectos como los que se mencionan a continuación:



Las pruebas unitarias se caracterizan porque se logran ejecutar si necesidad de la intervención manual. De igual modo, esta característica posibilita la automatización de su ejecución. Este tipo de pruebas se pueden repetir gran cantidad de veces. Por tal motivo, la rapidez de las pruebas es clave.

Otros factores que sobre las pruebas unitarias se deben tener presentes son:

* Las pruebas pueden cubrir casi la totalidad del código del programa desarrollado.
* Es una prueba excelente debido a su gran cobertura en el código.
* Las pruebas unitarias se deben poder ejecutar de manera independiente del estado al entorno de desarrollo.
* Se deben lograr ejecutar en cualquier ordenador del equipo de desarrollo.
* Al ejecutar este tipo de prueba, su ejecución no puede afectar la de otra prueba.
* Luego de la ejecución de la prueba, el entorno del desarrollo debe quedar igual que al inicio.
* Las relaciones entre los módulos deben ser simuladas para que no dependan de otros módulos.



La implementación de pruebas unitarias dentro de un equipo de desarrollo implica también una gran inversión de capacitación. En las primeras ejecuciones de las pruebas unitarias se tardará más tiempo en lograr ejecutarlas bien, ya que implica la familiarización de las herramientas con las cuales se ejecutan dichas pruebas.



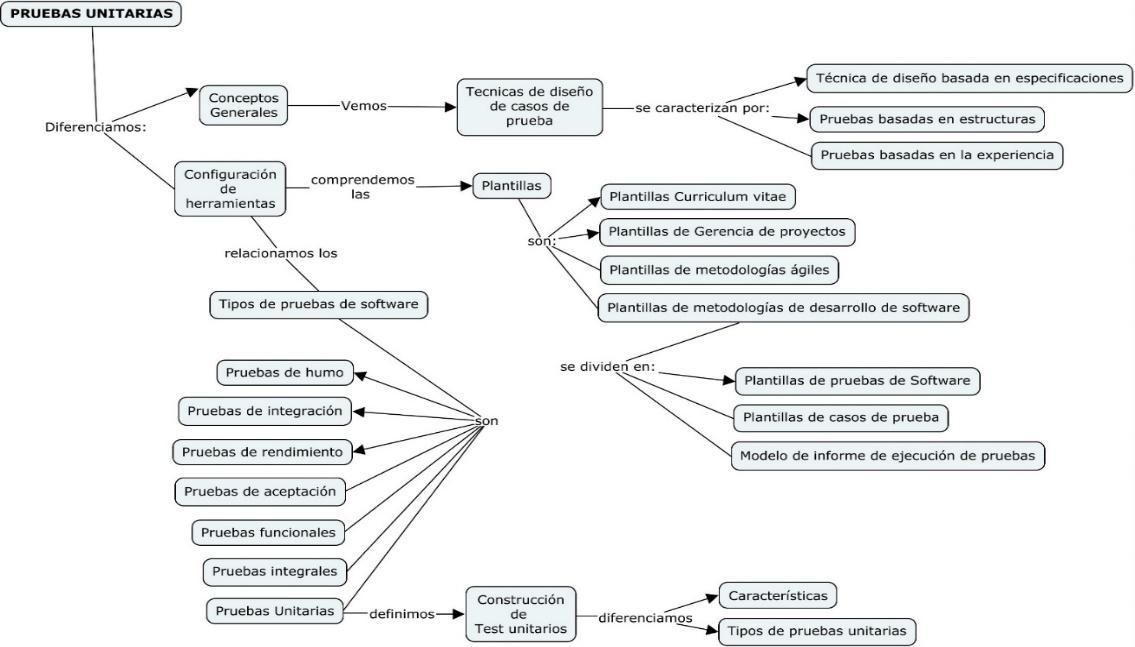
Analice el siguiente esquema y conozca más sobre los tipos de pruebas unitarias:





**Síntesis**

Aquí finaliza el estudio de las temáticas de este componente formativo. En este punto haga un análisis de la estructura que se muestra enseguida y registre su propia síntesis en su libreta personal de apuntes. Además, haga un repaso de los temas que considere necesarios.



1. **ACTIVIDADES DIDÁCTICAS (Se debe incorporar mínimo 1, máximo 2)**

| DESCRIPCIÓN DE LA ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la actividad | Creando pruebas unitarias de *software* |
| Objetivo de la actividad | Fortalecer los conceptos y habilidades de aplicación para pruebas unitarias de *software*, con base en los contenidos del componente formativo. |
| Tipo de actividad sugerida | Preguntas de falso - verdadero |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexos:  Actividad\_Didactica\_1 |

1. **MATERIAL COMPLEMENTARIO**

| Tema | Referencia APA del material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del recurso o  archivo del documento o material |
| --- | --- | --- | --- |
| Configuración de herramientas | Ramírez, S. (2020). *Instalar librería JUnit manualmente para Java Netbeans + prueba unitaria* [video]. YouTube. <https://www.youtube.com/watch?v=cMx8jsgmXII> | Video | <https://www.youtube.com/watch?v=cMx8jsgmXII> |
| Construcción de *test* unitarios | Miranda, M. (2015). *Pruebas unitarias con JUnit en Netbeans* [video]. YouTube. <https://www.youtube.com/watch?v=fyPfiEXt5i8&t=173s&ab_channel=MynorMiranda> | Video | <https://www.youtube.com/watch?v=fyPfiEXt5i8&t=173s&ab_channel=MynorMiranda> |

1. **GLOSARIO**

| TÉRMINO | SIGNIFICADO |
| --- | --- |
| *Development* | Es el conjunto de profesionales que desarrollan el producto con ciertas habilidades técnicas. |
| Diseño | Actividad creativa que proyecta objetos útiles. |
| Plantilla | Pieza que sirve de modelo o guía para ejecutar algo. |
| Prueba | Acción o efecto de probar cualquier especie de razón o medio. |

1. **REFERENCIAS BIBLIOGRÁFICAS**

Acervo Lima. (s.f.). *Entorno de prueba para pruebas de software*. Acervo Lima.

<https://es.acervolima.com/entorno-de-prueba-para-pruebas-de-software/>

Atlassian. (2022). *Los distintos tipos de pruebas en software.* Atlassian. <https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>

Quality Stream. (2020). *Cómo escribir casos de pruebas* *paso a paso* [video]. YouTube. <https://www.youtube.com/watch?v=laawNIdX9js&ab_channel=Quality-Stream>

Quality Stream. (2020). ¿*Qué necesitas saber para empezar como tester?* [video]. YouTube. <https://www.youtube.com/watch?v=vDhAgZMoRlM&ab_channel=Quality-Stream>

Ramírez, S. (2020). *Instalar librería JUnit manualmente para Java Netbeans + prueba unitaria* [video]. YouTube. <https://www.youtube.com/watch?v=cMx8jsgmXII&ab_channel=SANTIAGORAMIREZ>

Spanish Software Testing Qualifications Board. (2022). *Objetivos del SSTQB*. SSTQB. <https://es.sstqb.com>

1. **CONTROL DEL DOCUMENTO**

|  | Nombre | Cargo | Dependencia | Fecha |
| --- | --- | --- | --- | --- |
| Autor (es) | Luis Eyder Ortiz | Experto temático | Regional Norte de Santander – Centro de la Industria, la Empresa y los Servicios | Septiembre de 2022 |
| Javier Ricardo Luna Pineda | Diseñador instruccional | Regional Norte de Santander - Centro de la Industria, la Empresa y los Servicios | Septiembre de 2022 |
| Fabián Leonardo Correa Díaz | Diseñador instruccional | Regional Norte de Santander – Centro de la Industria, la Empresa y los Servicios | Octubre de 2022 |
| Andrés Felipe Velandia Espitia | Asesor metodológico | Regional Distrito Capital – Centro de Diseño y Metrología | Octubre de 2022 |
| Rafael Neftalí Lizcano Reyes | Responsable Equipo desarrollo curricular | Regional Santander – Centro Industrial del Diseño y la Manufactura | Octubre de 2022 |
| Julia Isabel Roberto | Correctora de estilo | Regional Distrito Capital – Centro de Diseño y Metrología | Octubre de 2022 |

1. **CONTROL DE CAMBIOS**

|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| --- | --- | --- | --- | --- | --- |
| Autor (es) |  |  |  |  |  |