



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Tecnólogo en desarrollo de medios gráficos visuales.
- **Código del Programa de Formación:** 522309.
- **Nombre del Proyecto :** Desarrollo de una estrategia de comunicación gráfica fundamentada en las necesidades del cliente del departamento del quindío
- **Fase del Proyecto :**
- **Actividad de Proyecto:** .
- **Competencia:** *Utilizar herramientas informáticas de acuerdo con las necesidades de manejo de información*
- **Resultados de Aprendizaje Alcanzar:**
 - 220501046-01- Alistar herramientas de tecnologías de la información y la comunicación (TIC), de acuerdo con las necesidades de procesamiento de información y comunicación.
 - 220501046-02- Aplicar funcionalidades de herramientas y servicios TIC, de acuerdo con manuales de uso, procedimientos establecidos y buenas prácticas.
 - 220501046-03- Evaluar los resultados, de acuerdo con los requerimientos.
 - 220501046-04- Optimizar los resultados, de acuerdo con la verificación.
- **Duración de la Guía:**

2. PRESENTACIÓN

El objetivo de la actividad es la identificación y apropiación de los conceptos sobre el hardware y software, su clasificación y las herramientas que ofrece internet.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

CN-GA-01-220501046-AA1 – Reconocer hardware, software de sistemas y software de aplicaciones.

El aprendiz de forma individual el aprendiz, realizara la definicion de los siguientes términos sobre los procesos Hardware, software según los siguientes interrogantes:

- **Términos generales**
 - Informática
 - Computadora / PC / Ordenador
 - Hardware
 - Software



- Ofimática
- Red / Red de cómputo / Red informática

▪ **Términos relacionados al HARDWARE**

- CPU
- Componentes internos
- Puertos
- Dispositivos de entrada o salida / Periféricos

▪ **Términos relacionados al SOFTWARE**

- firmware
- Sistema de inicio o BIOS
- Sistema operativo
- Programa informático
- Archivo
- Extensión
- Malware o virus
- Antimalware o Antivirus

▪ **Términos relacionados a las REDES**

- LAN
- WLAN / Wireless LAN
- WAN
- Internet
- Servidor
- Cliente
- Infraestructura de red

▪ **Términos relacionados a INTERNET**

- Web / WorldWide Web / “La red”
- La nube
- Servicio web
- Hipertexto
- Navegador
- Sitio web
- Almacenamiento en la nube
- Correo electrónico

- Identifica los elementos Externos (Periféricos) de una computadora.
- Realiza un cuadro donde describa las partes de las computadoras y periféricos como se describe a continuación:

Nombre de Parte o Periférico	Función	Imagen



--	--	--

- Identifica los elementos Internos de una computadora.
 - Realiza un cuadro donde describa las partes como se describe a continuación:

Nombre de Parte	Función	Imagen

- Identificar y clasifica los periféricos de entrada, Salida y Entrada y salida.
 - Realiza un cuadro donde describa las partes como se describe a continuación:

Nombre de Parte	Función	Imagen

- Identifica los puertos de periféricos mas comunes.
 - Realiza un cuadro donde enliste los puertos de interfaz de la computadora e indica ejemplos de dispositivos a conectar.

Nombre del Puerto	Imagen del Puerto	Dispositivo ejemplo a conectar
USB		Memorias USB, Cámaras Digitales, Impresoras.

- Identifica los tipos de Equipos de Computo sus caraterisrticas y utilidad.
- Identifica los tipos de sistemas operativos que existen.
- ¿Que son las Unidades de medida informática?
- ¿Identifica cuáles son las Unidades de almacenamiento?
- ¿Identifica cuáles son Las medidas de capacidad?

• EVIDENCIAS DE APRENDIZAJE.

Evidencia (De Conocimiento):

CN-GA-01-220501046-AA1-EV1– Reconocer hardware, software de sistemas y software de aplicaciones.

- Elementos para tener en cuenta en el documento:



- **Documento en el procesador de texto(word)** usando las normas Apa 7th edición, donde organice los puntos del proceso de conceptualización de la actividad. el documento debe tener como mínimo una portada, introducción, Tabla de contenidos, Desarrollo de los puntos de conceptualización, selección de herramientas de versionamiento a utilizar, conclusiones y referencias Bibliograficas.
- **Lineamientos generales para la entrega de la evidencia:**
 - **Productos para entregar:** Documento Reconocimiento de hardware, software de sistemas y software de aplicaciones.
 - **Formato:** PDF.
 - **Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio:** CN-GA-01-220501046-AA1-EV1- Reconocer hardware, software de sistemas y software de aplicaciones.

4. ACTIVIDADES DE EVALUACIÓN

Tome como referencia la técnica e instrumentos de evaluación citados en la guía de Desarrollo Curricular

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : Evidencias de Desempeño Evidencias de Producto:		Lista de chequeo Observacion directa. Lista de chequeo Valoracion de producto.

5. GLOSARIO DE TÉRMINOS

6. REFERENTES BIBLIOGRÁFICOS

Aula formativa. (2019, julio 29). ¿Qué es el Material Design?. [Web log post]. [blog.aulaformativa.com](https://blog.aulaformativa.com/que-es-el-material-design).
<https://blog.aulaformativa.com/que-es-el-material-design>

Beust, C. (2019). TestNG. <https://testng.org/doc>



Lázaro, G. (2015). Efecto Ripple de Material Design [HTML+CSS+JS]. <https://gersonlazarro.com/tutorial-efecto-ripple-de-material-design-html-css-js>

Material.io. (2021). Material Design. <https://material.io/design/color/the-color-system.html#color-usage-and-palettes>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Carlos Enrique Navia Torres	Instructor	Centro de Comercio y servicios.	19 enero 2024

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL
FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Tecnólogo en Análisis y Desarrollo de Software.
- **Código del Programa de Formación:** 228118.
- **Nombre del Proyecto:** Construcción de software integrador de tecnologías orientadas a servicios
- **Fase del Proyecto:** Ejecución.
- **Actividad de Proyecto:** .
- **Competencia:** Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.
- **Resultados de Aprendizaje Alcanzar:**
 - Crear componentes front-end del software de acuerdo con el diseño
- **Duración de la Guía:** 6 horas

2. PRESENTACIÓN

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

CN-GA-01-AA1 Establecer los componentes frontend de la aplicación web.

- Descripción de la(s) Actividad(es):

CN-GA1-01-AA1 – EV01 Gestion de dependencias de código según lenguaje de programación seleccionado(Javascript).

El aprendiz de forma individual el aprendiz, realizara la definicion de los siguientes términos sobre los procesos de gestion de dependencias en Javascript según los siguientes interrogantes:

- ¿Que es un gestor dependencias de codigo?
- ¿Qué es npm?
- ¿Para qué se utiliza principalmente npm?
- ¿Qué es el versionado semántico?
- Como está especificado el Versionado Semántico.
- Qué son las dependencias locales
- Qué son las dependencias de desarrollo
- Qué son las dependencias globales
- ¿Que es el archivo `package.json` y que apartados tiene y cual es su utilidad?



- ¿Qué es el archivo *package-lock.json* y que utilidad tiene?
- que es la carpeta *node_modules* en un proyecto de npm
- Realiza una **Cheat sheet** de los siguientes procesos (ver ejemplo: <https://pin.it/2TauqeogV>):
 - como inicializar un proyecto de npm.
 - Cómo instalar dependencias locales
 - Cómo instalar dependencias de desarrollo
 - Cómo instalar dependencias globales
 - Cómo visualizar las dependencias instaladas
 - Cómo instalar una versión específica de un paquete
 - Cómo crear un comando en tu proyecto (scripts en el `package.json`)
 - Cómo actualizar dependencias
 - Cómo eliminar paquetes
 - Cómo actualizar *node_modules*

- **EVIDENCIAS DE APRENDIZAJE.**

Evidencia (De Conocimiento):

- **CN-GA1-01-AA1-EV01** Gestion de dependencias de código según lenguaje de programación seleccionado (Javascript).
 - **Elementos para tener en cuenta en el documento:**
 - **Documento en el procesador de texto (word)** usando las normas Apa 7th edición, donde organice los puntos del proceso de conceptualización de la actividad. el documento debe tener como mínimo una portada, introducción, Tabla de contenidos, Desarrollo de los puntos de conceptualización, selección de herramientas de versionamiento a utilizar, conclusiones y referencias Bibliográficas.
 - **Lineamientos generales para la entrega de la evidencia:**
 - **Productos para entregar:** Documento informe técnico
 - **Formato:** PDF.
 - **Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio:** CN-GA1-01-AA1-EV01 Gestion de dependencias de código según lenguaje de programación seleccionado (Javascript).

Evidencia (De Producto):

- **Elementos para tener en cuenta en el documento:**



- **Cheat sheet:** Donde describa los procesos de gestion de pendencias.

- **Lineamientos generales para la entrega de la evidencia:**
 - **Productos para entregar:** Cheat sheet
 - **Formato:** PDF.
 - **Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio:** CN-GA1-01-AA1-EV01 Gestion de dependencias de código según lenguaje de programación seleccionado(Javascript).

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencia de conocimiento: CN-GA1-01-AA1-EV01 Gestion de dependencias de código según lenguaje de programación seleccionado(Javascript). Evidencia de Desempeño: Evidencias de Producto:	<ul style="list-style-type: none">• Crea interfaces de usuario aplicando buenas prácticas de usabilidad y accesibilidad.• Genera plantillas y estilos de acuerdo con el diseño establecido.	IE- CN-GA1-01-AA1-EV01 Lista de chequeo

5. GLOSARIO DE TÉRMINOS

- **npm (Node Package Manager):** Un gestor de paquetes para JavaScript, utilizado para instalar, compartir y administrar bibliotecas y dependencias en proyectos de JavaScript.
- **package.json:** Un archivo en formato JSON presente en la raíz de un proyecto Node.js. Define las propiedades del proyecto, incluyendo metadatos, dependencias, scripts, y más.
- **Node.js:** Un entorno de ejecución para JavaScript en el servidor, basado en el motor V8 de Chrome.
- **Dependencia:** Un paquete o módulo que un proyecto necesita para funcionar correctamente. Las dependencias se instalan y gestionan a través de npm.
- **devDependencies:** Dependencias necesarias solo durante el desarrollo del proyecto, como herramientas de compilación o pruebas.
- **Global Installation:** La instalación de un paquete de npm de manera global en el sistema, lo que permite su uso en cualquier proyecto en la máquina.



- **Local Installation:** La instalación de un paquete de npm específicamente en un proyecto. El paquete se encuentra en la carpeta `node_modules` del proyecto.
- **node_modules:** Una carpeta en un proyecto Node.js donde se almacenan las dependencias locales instaladas a través de npm.
- **Semantic Versioning (SemVer):** Un sistema de versionamiento para paquetes, que indica las versiones en formato `mayor.menor.patch`, gestionando compatibilidad y nuevas características.
- **npm CLI (Command Line Interface):** La interfaz de línea de comandos de npm, utilizada para ejecutar comandos de npm, como instalar o actualizar paquetes.
- **npm Registry:** Un repositorio público de paquetes de npm, donde los desarrolladores pueden publicar y compartir paquetes de JavaScript.
- **npx:** Una herramienta de npm que permite ejecutar paquetes npm sin instalarlos globalmente.
- **Scripts:** Comandos personalizados definidos en el archivo `package.json` que se pueden ejecutar a través de npm para automatizar ciertas tareas.
- **npm audit:** Una funcionalidad de npm que analiza las dependencias del proyecto en busca de vulnerabilidades de seguridad conocidas.
- **package-lock.json:** Un archivo generado automáticamente por npm que registra la versión exacta de cada paquete instalado, asegurando la consistencia entre instalaciones.

6. REFERENTES BIBLIOGRÁFICOS

- Node.js Foundation. (2023). Node.js User Guide. Node.js Foundation. <https://nodejs.org/en/docs/>
- NPM, Inc. (2023). NPM Documentation. <https://docs.npmjs.com/>
- Richards, J. (2022). Mastering Node.js: Building Scalable and Performant Applications. O'Reilly Media.
- Thompson, H., & Patel, R. (2021). JavaScript and Node.js: Full Stack Development and Applications. Cambridge University Press.
- Green, L. (2020). Effective JavaScript Package Management with npm. Addison-Wesley.
- Patel, S. (2023). "Understanding Semantic Versioning for JavaScript Developers", Journal of Web Development, 45(2), 134-145.
- Jenkins, M., & O'Connor, T. (2021). Security in Node.js and npm: Managing Dependencies and Deployments. Apress.
- Warner, A. (2019). "The Evolution of npm and Node.js: A Look at Modern Web Development", Journal of Software Engineering, 12(3), 201-215.



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Carlos Enrique Navia Torres	Instructor	CCyT	22/01/2024

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Tecnólogo en Análisis y Desarrollo de Software.
- **Código del Programa de Formación:** 228118.
- **Nombre del Proyecto:** Construcción de software integrador de tecnologías orientadas a servicios
- **Fase del Proyecto:** Ejecucion.
- **Actividad de Proyecto:** .
- **Competencia:** 220501096 - Construccion del Software
- **Resultados de Aprendizaje Alcanzar:**
 - 220501096-01 Planear actividades de construcción del software de acuerdo con el diseño establecido.
- **Duración de la Guía:** 40 horas

2. PRESENTACIÓN

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

CN-GA1-220501096-01-AA1 configurar herramientas de versionamiento y planificación para control de código.

- Descripción de la(s) Actividad(es):

CN-GA1-220501096-01-AA1 – EV01 Informe técnico de plan de trabajo para construcción de software.

El aprendiz de forma individual el aprendiz, realizara la definicion de los siguientes términos sobre los procesos devOps e integracion continua según los siguientes interrogantes:

A. Conceptos de DevOps e Integracion Continua.

- ¿Qué es DevOps?
- ¿Cómo funciona DevOps?
- Identifica El ciclo de vida de DevOps y describa la función de cada ciclo.
- ¿Cuáles son las ventajas de DevOps?
- ¿Cuáles son los desafíos de adoptar DevOps?
- ¿Que es la integración continua CI?
- Que es el vesionado semantico
- Como está especificado del Versionado Semántico, indique 3 ejemplos de como hacerlo.



B. Herramientas de DevOps

○ *Sistemas de control de versiones.*

Defina los siguientes términos con su equipo de trabajo sobre los sistemas de control de versiones:

- Que es un sistema de control de versiones en el desarrollo de software.
- Características que tiene un sistema de control de versiones.
- Identifica los diferentes tipos de sistemas de control de versiones que existen en el mercado, **hacer una tabla comparativa.**
- Identifica las plataforma web que proporciona servicios de alojamiento de repositorios de control de versiones y colaboración en software, **hacer presentación con tabla comparativa.**
- Identificar los clientes visuales para gestionar el control de versiones, **hacer presentación con tabla comparativa.**

○ *Herramientas orquestadoras de Integración Continua y Entrega Continua (CI/CD).*

- Identifica los herramientas Orquestadoras de CI/CD que existen en el mercado, **hacer una tabla comparativa.**

○ *Herramientas de Configuración y Gestión de Infraestructura.*

- Identifica los herramientas Configuración y Gestión de Infraestructura que existen en el mercado, **hacer una tabla comparativa.**

○ *Herramientas de Colaboración y Gestión de Proyectos.*

- Identifica las Herramientas que ayudan en la colaboración entre equipos y en la gestión de proyectos. Hacer una tabla comparativa.

○ *Herramientas de Monitorización y Registro.*

- Identifica los herramientas de Monitorización y Registro que existen en el mercado, **hacer una tabla comparativa.**

○ *Herramientas de Contenedores y Orquestación.*

- Identifica los herramientas Contenedores y Orquestación que existen en el mercado, **hacer una tabla comparativa.**

○ *Herramientas de Seguridad DevOps (DevSecOps).*

- Identifica los herramientas Seguridad DevOps (DevSecOps) que existen en el mercado, **hacer una tabla comparativa.**

○ *Herramientas de Automatización de Pruebas.*

- Identifica los herramientas Automatización de Pruebas que existen en el mercado, **hacer una tabla comparativa.**

Nota importante: Minimo se debe relacionar 3 herramientas por cada categoría de las herramientas.



C. Realizar informe Técnico de selección de herramientas a utilizar en el proyecto definido.

- En el documento de trabajo cree un apartado donde Realice un informe técnico especificando herramientas y tecnologías de versionamiento a utilizar según lo visto en la revisión de Herramientas de DevOps que aplicaría en el proyecto que tenga en mente.

• EVIDENCIAS DE APRENDIZAJE.

Evidencia (De Conocimiento):

- **CN-GA1-220501096-01-AA1–EV01 Informe técnico de plan de trabajo para construcción de software.**

- **Elementos para tener en cuenta en el documento:**

- Documento en el procesador de texto(word) usando las normas Apa 7th edición, donde organice los puntos del proceso de conceptualización de la actividad. el documento debe tener como mínimo una portada, introducción, Tabla de contenidos, Desarrollo de los puntos de conceptualización, selección de herramientas de versionamiento a utilizar, conclusiones y referencias Bibliograficas.

- **Lineamientos generales para la entrega de la evidencia:**

- **Productos para entregar:** Documento informe técnico
- **Formato:** PDF.
- **Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio:** Informe técnico de plan de trabajo para construcción de software CN-GA1-220501096-01-AA1-EV01.

Evidencia (De Producto):

Evidencia (De desempeño)



4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencia de conocimiento: CN-GA1-220501096-01-AA1-EV01 Informe técnico de plan de trabajo para construcción de software. Evidencia de Desempeño: Evidencias de Producto:	<ul style="list-style-type: none">• Interpreta el informe de diseño para definir el plan de trabajo en la construcción del software.• Selecciona y configura herramientas de desarrollo de acuerdo con las condiciones del software a construir.• Define estándares de codificación de acuerdo con las reglas de la plataforma de desarrollo seleccionada.	IE-GA7-220501096-AA1-EV01 Lista de chequeo

5. GLOSARIO DE TÉRMINOS

- **APA (American Psychological Association):** La APA es una organización profesional que establece estándares para la comunicación científica y académica, incluyendo la escritura, citación y formato de documentos.
- **Cita:** Una cita es una referencia a una fuente de información específica en un trabajo académico para dar crédito al autor original y proporcionar evidencia para respaldar afirmaciones.
- **Fuente secundaria:** Una fuente secundaria es un trabajo que cita o hace referencia a una fuente primaria, por lo que no se basa directamente en la fuente original.
- **Integración Continua (CI):** Práctica de automatizar la integración de cambios de código de varios contribuyentes en un único proyecto de software de manera frecuente, generalmente varias veces al día.
- **Pipeline de CI/CD:** Conjunto de pasos automatizados que se utilizan para llevar el código desde el desarrollo hasta la producción. CI se refiere a la integración continua de cambios en el código y CD puede referirse a la entrega continua o despliegue continuo.
- **Control de Versiones:** Sistema que registra los cambios en un archivo o conjunto de archivos a lo largo del tiempo para que puedas recuperar versiones específicas más tarde. Ejemplos incluyen Git y SVN.
- **Repositorio:** Lugar donde se almacena el código fuente y se realiza el seguimiento de su historial de cambios, a menudo utilizando sistemas de control de versiones.



- **Commit:** Acción de guardar cambios en el sistema de control de versiones. Un commit representa una unidad atómica de cambio y lleva consigo un mensaje descriptivo.
- **Branch (Rama):** Versión paralela del código fuente, derivada del mismo, que permite trabajar en cambios o características sin afectar al código principal.
- **Merge (Fusión):** Proceso de integrar cambios desde una rama a otra, comúnmente de una rama de desarrollo a la rama principal.
- **Build (Construcción):** Proceso automatizado de convertir el código fuente en un paquete ejecutable.
- **Pruebas Automatizadas:** Tests que se ejecutan automáticamente para asegurar que el código cumple con los requisitos previstos y no introduce errores.
- **Servidor de CI:** Servidor que orquesta la ejecución automática de los pipelines de CI/CD. Ejemplos incluyen Jenkins, Travis CI y CircleCI.
- **Docker/Contenedores:** Tecnología que permite empaquetar y aislar aplicaciones con todo su entorno. Muy utilizada para garantizar la coherencia entre entornos de desarrollo, pruebas y producción.
- **Despliegue Continuo (CD):** Extensión de la entrega continua que automatiza el despliegue de aplicaciones en entornos de producción.
- **Entrega Continua (CD):** Práctica de automatizar el proceso de entrega de software de tal manera que sea seguro liberar una nueva versión en cualquier momento.
- **Rollback:** Proceso de revertir a una versión anterior del software si la nueva versión falla o introduce errores.
- **Integración de Terceros:** Inclusión y manejo de servicios, APIs o bibliotecas desarrolladas por terceros en el proceso de CI/CD.
-

6. REFERENTES BIBLIOGRÁFICOS

- American Psychological Association (2020). Publication manual of the American Psychological Association (7th ed.). <https://doi.org/10.1037/0000165-000>
- American Psychological Association (01 de enero de 2020). Style and Grammar Guidelines. Recuperado el 17 de enero de 2020 de <https://apastyle.apa.org/style-grammar-guidelines/index>
- Atlassian. (s.f.). Qué es el control de versiones | Atlassian Git Tutorial. <https://www.atlassian.com/es/Git/tutorials/what-is-version-control>
- Git. (2021) Git - Acerca del control de versiones. Git --Local-Branching-on-the-Cheap. <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>
- Younes. (2021). Gitlab VS Github VS BitBucket. Which one deserve your time? DEV Community. <https://dev.to/yafkari/gitlab-vs-github-vs-bitbucket-which-one-deserve-your-time-2npm>



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Carlos Enrique Navia Torres	Instructor	CCyT	22/01/2024

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					