



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

Guía Proceso de Inducción aprendices

- Denominación del Programa de Formación: Curso Introductorio a la Formación Profesional Integral
- Código del Programa de Formación: 41311582
- Nombre del Proyecto Formativo (si aplica):
- Fase del Proyecto (si aplica):
- Actividad de Proyecto Formativo (si aplica):
- Competencia: Resultado de Aprendizaje de la Inducción.
- Resultados de Aprendizaje: Identificar la dinámica organizacional del SENA y el rol de la formación profesional integral de acuerdo con su proyecto de vida y el desarrollo profesional
- Duración de la Guía de Aprendizaje: Cuarenta (40) Horas

2. PRESENTACIÓN

En el marco del Programa de Articulación del SENA con la Educación Media, liderado por el Gobierno Nacional a través del SENA y el Ministerio de Educación Nacional para estudiantes, ustedes hacen parte de una estrategia que les permite obtener doble titulación: Bachiller y Técnico SENA, fortaleciendo sus competencias laborales y ampliando sus oportunidades de inserción en el sector productivo. Este programa contribuye al cierre de brechas en el acceso a la formación técnica y al empleo, promoviendo la continuidad educativa y la proyección



hacia niveles tecnológicos y profesionales, gracias a los procesos de homologación con instituciones de educación superior.

Esta formación técnica, desarrollada directamente en su institución educativa, responde a las necesidades

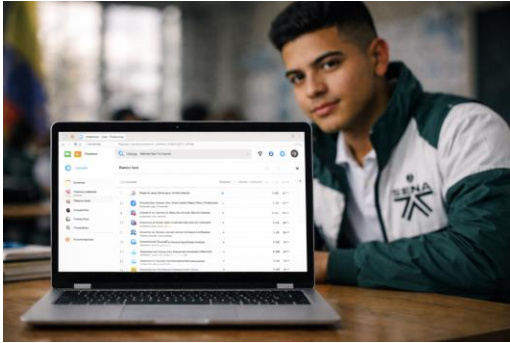
del contexto productivo colombiano y les brinda herramientas reales para su futuro laboral, académico y social. Aprovechen cada actividad como una oportunidad para construir conocimiento, fortalecer sus competencias y aportar al desarrollo económico y social del país.

¡Bienvenidos a este proceso de formación profesional integral que transforma saberes en oportunidades y aprendizajes en proyectos de vida!

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

Las siguientes actividades lo Orientarán en la dinámica y fortalecimiento del sentido de pertenencia de su institución formadora el SENA y en el proceso formativo que realizará a continuación:

Antes de empezar lo invitamos a ver la bienvenida del actual director general del SENA, dando clic en el siguiente enlace <https://www.youtube.com/watch?v=NOJpxIH0a6E> o accediendo al material de apoyo de la guía de inducción “Bienvenida Aprendices 2025 – SENA.mp4”.



Portafolio del aprendiz

Ahora vamos a crear el portafolio del aprendiz, donde podrá recopilar y almacenar las evidencias de su proceso formación. Si ya cuenta con correo @soy.sena.edu.co, podrá usar el espacio del one drive para esta tarea, de lo contrario se recomienda crear una cuenta de correo (Gmail) nueva y de uso

exclusivo para su portafolio SENA. La estructura del portafolio la encontrará en el material de apoyo como: **“Portafolio_Aprendiz.png”**

Base de Datos de aprendices

Por último, no olvide suministrar la información solicitada por su instructor para conformar la base de datos de su grupo o ficha. Para mayor agilidad puede diligenciar el siguiente link.

(poner Link propio)

Ahora sí, estamos listos para iniciar. Para el desarrollo de este proceso introductorio a la formación profesional integral, se proponen cuatro (4), actividades de aprendizaje, las cuales se relacionan a continuación. **¡Éxitos!**

Elección Vocero

Es un momento clave en el que un aprendiz apoye las actividades en el proceso de formación, debe ser un proceso democrático mediante el cual los aprendices del grupo eligen a un representante que actúa como puente entre el grupo, los instructores.

Las funciones principales de los voceros:

- Comunicar inquietudes, propuestas y necesidades del grupo.
- Apoyar la organización de actividades académicas.
- Promover la convivencia, el respeto y el cumplimiento del reglamento del aprendiz.
- Participar en reuniones cuando sea convocado.

Responsabilidad y liderazgo. buena comunicación, respeto por compañeros e instructores, Compromiso con la formación, actitud propositiva y conciliadora.



Ahora identifique su estilo de aprendizaje

realizando el “**Test de David Kolb.xlsx**” (Excel) el cual encontrará en la carpeta material de formación y guárdela con el nombre de “Mi estilo de aprendizaje.xlsx”.

Ambiente requerido: **Aula Convencional**

Estrategias o técnicas didácticas activas: Reflexión individual guiada, Aprendizaje autónomo Autodiagnóstico, Análisis de contenido audiovisual, Uso pedagógico de TIC Socialización voluntaria de resultados

Materiales de formación: Computador o dispositivo móvil, Acceso a internet, Herramientas ofimáticas (Word, Excel)

Material de apoyo:

- Documento: *Valores, defectos y cualidades.pdf*
- Video: *KOLB y los estilos de aprendizaje.mp4*
- Archivo: *Test de David Kolb.xlsx*
- Enlaces web sugeridos en la guía

Evidencias:

- Imagen “Quien soy.jpg”.
- Hoja de cálculo “Mi estilo de aprendizaje.xlsx”

Duración de la actividad: horas.

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

Contextualización Senalogía

A continuación, conoceremos la Historia, misión y visión del SENA. Sus símbolos Institucionales y plataformas como parte integral de mi identidad y sentido de pertenencia como aprendiz SENA.

Ahora que somos parte del SENA debemos reconocer las características que nos identifican como



aprendiz de esta prestigiosa institución. De la misma manera que nuestro colegio tiene un escudo, una bandera y un uniforme característico, el SENA también tiene unos símbolos institucionales, una misión y visión que debemos conocer.

Historia, misión y visión y Símbolos Institucionales del SENA.

Descripción de la actividad: Diríjase al material de apoyo donde encontrará la carpeta “Identidad SENA” o si lo prefiere haga clic en el siguiente enlace <https://www.sena.edu.co/es-co/sena/Paginas/quienesSomos.aspx> Luego de leer y explorar el material, realice una infografía

donde se muestre la historia, la misión, visión y los símbolos institucionales del SENA y guárdela con el nombre “**Info_Identidad SENA.pdf**”. ¿Sabía que el SENA tiene un himno? Conózcalo aquí <https://share.google/uLXKDvw53AFqCR7xg>

Plataformas SENA.

Descripción de la actividad: Nuestra formación es presencial, pero nos apoyamos en algunas



plataformas tecnológicas que nos permiten administrar, organizar y acceder de manera eficiente a toda la información entorno al proceso de formación. Ingrese a cada una de las plataformas que se mencionan a continuación y explore cada rincón, tome capturas de pantalla (mínimo 5 por cada

plataforma) insértelas en un documento de Word y agregue una descripción de cada captura.

Guarde el documento como “**Plataformas SENA.pdf**”. Pídale a su instructor que le sugiera un curso complementario virtual e inscríbase.



- **SofiaPlus** <https://oferta.senasofiaplus.edu.co/sofia-oferta/>

Actualización de datos

Descarga de certificados de estudios

- **ZAJUNA** <https://zajuna.sena.edu.co/>

Cursos virtuales

Enlace de PQRS

Soporte técnico

- **Sistemas Bibliotecas** <https://biblioteca.sena.edu.co/>

Repositorio

- **Betowa** <https://betowa.sena.edu.co/oferta>

Consultar programas de formación

- **Blog CGMLTI** <https://mercadoslogisticaytecnologia.blogspot.com/>

Beneficios del Centro de formación

Programas de formación

Subdirector y coordinación

También usamos Google DRIVE y MS TEAMS.

Ambiente requerido: **Aula Convencional**

Estrategias o técnicas didácticas activas: Aprendizaje por exploración, Análisis de información institucional, Aprendizaje significativo, Uso de TIC, Síntesis visual (infografía)

Materiales de formación: Computador con acceso a internet, Herramientas de diseño (Canva, PowerPoint, Word), Navegador web, Plataformas institucionales del SENA

Material de apoyo: Carpeta: Identidad SENA, Página web oficial del SENA, Video del Himno del SENA

Enlaces a plataformas: Sofía Plus, Zajuna, Biblioteca SENA, Betowa, Blog CGMLTI

Evidencias



Infografía Identidad del SENA

Pantallazos de plataformas institucionales

Duración de la actividad: horas.

3.3 Actividades de apropiación:

Comprender en qué consiste la formación por proyectos como estrategia para el logro de los resultados del programa, así como la normas que rigen la formación.

¿Ha notado que el SENA maneja un lenguaje diferente para nombrar algunos conceptos muy parecidos a lo que vivimos a diario en el colegio? Palabras como: Formación, Instructor, Ambiente de aprendizaje, Evidencias de aprendizaje y Resultados de aprendizaje entre otros. Esto es porque estamos hablando de formación Técnica, veamos de que se trata desarrollando las siguientes actividades.

FPI, Formación por proyectos, Programa de formación.

Descripción de la actividad: Siga la presentación “FPI Modelo Pedagógico SENA.pptx” mientras escucha atentamente a su instructor quien explicará en qué consiste la Formación Profesional Integral FPI, la formación por proyectos del SENA y las Guías que se trabajaran en cada grado. Una vez terminada la explicación busque en el material de apoyo el programa de Formación y/o Diseño curricular y realice una presentación en PowerPoint u otra herramienta de su preferencia (Canva, Prezi, PowToon) sintetizando los siguientes puntos del programa de formación:

- **Perfil idóneo de egreso**
- **Perfil ocupacional**
- **Perfil profesional**
- **Proyección del egresado**
- **Descripción general del perfil del egresado**



CONTENIDOS CURRICULARES DE LA COMPETENCIA COMPETENCIAS

(Solo de las competencias Técnicas, consulte con su instructor)

- **NORMA / UNIDAD DE COMPETENCIA**
- **RESULTADOS DE APRENDIZAJE**
- **CONOCIMIENTOS, Procesos y criterios de Evaluación**

Reglamento del aprendiz.

Descripción de la actividad: Revise y Lea el documento “**Acuerdo_sena_0009_2024.pdf**” que trata sobre el reglamento del aprendiz SENA, allí encontrará los deberes y derechos del aprendiz entre otros temas de interés. Este documento se encuentra en la carpeta material de apoyo. Ahora utilizando la herramienta **ChatPDF** <https://www.chatpdf.com/es> o la herramienta Notebook Im de Google <https://notebooklm.google.com/> realice las siguientes preguntas sobre el manual del aprendiz SENA guarde el chat como “**Chat Reglamento Aprendiz.pdf**”.

1. En qué casos se considera que un aprendiz se encuentra en deserción.
2. Enumere 5 derechos del aprendiz.
3. Cuáles son los deberes más relevantes del aprendiz.
4. Que sucede si no asisto a la formación.
5. Que es un llamado de atención.
6. Que sucede si me cambio de colegio.

Realice 3 preguntas más de su interés.

Ambiente requerido: **Aula Convencional**

Estrategias o técnicas didácticas activas: Aprendizaje orientado por el instructor, Análisis documental
Aprendizaje colaborativo, Síntesis de información, Uso de herramientas de IA con fines académicos
Presentación de resultados.

Materiales de formación: Computador o dispositivo móvil, Presentación: FPI Modelo Pedagógico
SENA.pptx, Herramientas de presentación (PowerPoint, Canva, Prezi), Acceso a ChatPDF o NotebookLM



Material de apoyo: Diseño curricular del programa de formación, Documento: Acuerdo_sena_0009_2024.pdf, Guía de aprendizaje Enlaces a herramientas digitales sugeridas

Evidencias de aprendizaje:

Presentación del programa de formación

Chat generado sobre el reglamento: *Chat Reglamento Aprendiz.pdf*

Instrumentos de evaluación: Lista de Chequeo Reglamento del Aprendiz

Duración de la actividad: horas.

3.4 Actividades de Transferencia el Conocimiento:

A partir del proyecto formativo se debe proponer un **proyecto productivo** de acuerdo a unas líneas temáticas y a unos entregables preestablecidos que le permitirán poner en práctica las competencias adquiridas durante su proceso de formación.

Etapas productiva y proyecto productivo

Descripción de la actividad: Observe atentamente el video **“Etapas Productivas.mp4”** que se encuentra en el material de apoyo, luego escuche atentamente a su instructor quien explicará en qué consiste el proyecto formativo y las líneas de proyecto establecidas sobre las que podrá desarrollar la etapa práctica.

Por último, en grupos de 4 a 5 aprendices, seleccionen una línea o temática de proyecto y realice una lluvia de ideas y/o propuesta para la aplicación del proyecto productivo, genere un documento PDF y nómbrelo como **“Propuesta proyecto.pdf”**

Cadena de formación

¿Sabía que puede continuar con su carrera, ahorrando tiempo y dinero? La cadena de formación es la continuación a un nivel superior, previo el reconocimiento del ciclo anterior debidamente certificado, en el cual se habilita al aprendiz para el sector productivo, continuando el proceso de



desarrollo humano en la misma línea de ocupación, oficio o tecnología.

La cadena de formación le permitirá pasar del ciclo técnico al tecnológico y luego al profesional.

Descripción de la actividad: Consulte con su instructor cuales son las carreras tecnológicas y profesionales afines a su programa técnico, luego realice una línea de tiempo donde se proyecte como profesional indicando el año de finalización, título obtenido y centro de formación y/o universidad.

Guarde su evidencia como **“Línea de tiempo Profesional.png”**. En el siguiente enlace podrá encontrar algunos convenios con universidades e instituciones de educación superior.

https://www.sena.edu.co/es-co/comunidades/instructores/Paginas/convenios_con_universidades.aspx

Ambiente requerido: **Aula Convencional**

Estrategias o técnicas didácticas activas: Aprendizaje basado en proyectos, Trabajo colaborativo, Lluvia de ideas, Planeación y proyección profesional, Aprendizaje significativo, Transferencia del conocimiento, Materiales de formación,

Material de apoyo:

Evidencias de aprendizaje: Rúbrica – Proyecto Productivo

Documento: *Propuesta proyecto.pdf*

Línea de tiempo profesional: *Línea de tiempo Profesional.png*

Instrumentos de evaluación:

Duración de la actividad: horas.

GFPI-F-135 V04



	N/A	<p>como la normas que rigen la formación.</p> <p>Proponer ideas de proyecto de acuerdo con el proceso de etapa productiva del programa de formación</p>	<p>Documento PDF</p> <p>“Propuesta Proyecto.pdf”</p> <p>Imagen “Línea de tiempo Profesional.png”</p>		<p>Rúbrica – Proyecto Productivo</p> <p>Lista de Chequeo – Línea de Tiempo Profesional</p>
--	-----	---	--	--	--

5. GLOSARIO DE TÉRMINOS

Actividades de aprendizaje: Acciones realizadas por los aprendices con la orientación del instructor que son unos ejemplos de lo que se debe hacer en el mundo laboral.

Ambiente de aprendizaje: Espacio en el que converge el conjunto articulado de fuentes de conocimiento para desarrollar en el aprendiz competencias en el ámbito de la conciencia y la capacidad tecnológica, la capacidad de abstracción y la habilidad de adaptación a los cambios de las estructuras productivas. Se distinguen los siguientes tipos de ambientes: el ambiente polivalente, el ambiente pluritecnológico, la unidad productiva agropecuaria, el aula abierta de informática, el aula convencional, el aula móvil, el laboratorio, el auditorio, la biblioteca, el campo deportivo, el ambiente virtual y los ambientes fuera de Centro permiten dar un juicio durante y al final del proceso

Etapa Lectiva: Período en el cual el alumno recibe formación en ambientes educativos institucionales que Facilitan el logro de los objetivos de la formación.

Etapa Productiva: Tiempo en el que el alumno aplica los conocimientos adquiridos en la etapa lectivo, en situaciones reales de trabajo, donde la empresa participa activamente con el acompañamiento y asesoría necesarias para alcanzar los objetivos establecidos.



Evaluación del Aprendizaje: Son los juicios que se emiten con base en los resultados del aprendizaje y los criterios de evaluación que contribuyen a fortalecer el desarrollo del aprendiz.

Evidencias de Aprendizaje: Pruebas de aprendizaje recolectados por el instructor que permiten reconocer los logros del aprendiz en términos de saber, hacer y actuar.

Evidencias de conocimiento: Son las manifestaciones ciertas y claras sobre el aprendizaje que se relacionan con el saber, conocer de hechos y procesos, comprensión de los principios y teorías. Saber utilizar y aplicar el conocimiento.

Formación Profesional Integral: Proceso teórico-práctico que se orienta al desarrollo de conocimientos técnicos, tecnológicos de actitudes y valores para su convivencia social y laboral.

Guía de Aprendizaje: Recurso didáctico para facilitar el aprendizaje, se centra en la pedagogía activa y participativa. Incluye actividades planeadas por procesos diseñados por el instructor que promueve la comprensión y el conocimiento.

Modalidad de Formación: Forma de entender la formación profesional, esta parte puede ser presencial o virtual.

Programa de Formación: Conjunto de competencias de la certificación de la formación profesional integral para dar respuesta a las necesidades del sector productivo.

Resultados de Aprendizaje: Indicadores que orientan y permiten al instructor y al alumno la verificación de procesos requeridos para el logro de los objetivos del aprendizaje propuestos.

Criterios de Evaluación: Indicadores del nivel logrado por los alumnos en el proceso de aprendizaje que

6. REFERENTES BIBLIOGRÁFICOS

Servicio Nacional de Aprendizaje – SENA. (2024). Reglamento del Aprendiz SENA. Acuerdo 0009 de 2024. SENA.



Servicio Nacional de Aprendizaje – SENA. (s. f.). Modelo Pedagógico de la Formación Profesional Integral – FPI. SENA.

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall.

Ministerio de Educación Nacional. (2016). *Articulación de la Educación Media con la Educación Superior y la Formación para el Trabajo*. MEN.

Acuerdo 008 de 1997 estatutos de la formación profesional Integral

Manual para la articulación del SENA con la educación media - GFPI-M-004

Servicio Nacional de Aprendizaje – SENA.
¿Quiénes somos? Historia, misión y visión del SENA
<https://www.sena.edu.co/es-co/sena/Paginas/quienesSomos.aspx>

Servicio Nacional de Aprendizaje – SENA.
Plataforma SOFIA Plus
<https://oferta.senasofiaplus.edu.co/sofia-oferta/>

Servicio Nacional de Aprendizaje – SENA.
Plataforma ZAJUNA – Formación Virtual
<https://zajuna.sena.edu.co/>

Servicio Nacional de Aprendizaje – SENA.
Sistema de Bibliotecas SENA
<https://biblioteca.sena.edu.co/>

Servicio Nacional de Aprendizaje – SENA.
Plataforma BETOWA – Oferta de Programas
<https://betowa.sena.edu.co/oferta>

Servicio Nacional de Aprendizaje – SENA.
Convenios con Instituciones de Educación Superior
https://www.sena.edu.co/es-co/comunidades/instructores/Paginas/convenios_con_universidades.aspx

Ejemplos.co.
Cualidades y defectos personales



<https://www.ejemplos.co/40-ejemplos-de-cualidades-y-defectos/>

YouTube – Canal Oficial SENA.

Bienvenida aprendices SENA

<https://www.youtube.com/watch?v=NOJpxIH0a6E>

YouTube.

Estilos de aprendizaje – David Kolb

<https://www.youtube.com/watch?v=DnCaddP9f9M>

ChatPDF.

Herramienta de análisis de documentos PDF

<https://www.chatpdf.com/es>

Google.

NotebookLM – Análisis de documentos con IA

<https://notebooklm.google.com/>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Edwin Cañon	Instructor	Articulación	05 de febrero de 2026

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Julian Camilo Maturana	Instructor	Articulación	09 de febrero de 2026	Correcciones



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL
FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Técnico en Programación de Software
- Código del Programa de Formación: 233104 ver 2
- Nombre del Proyecto Formativo: Construcción de sistema de información para emprendedores y empresarios.
- Fase del Proyecto: Diseño o Formulación del Proyecto (Planeación).
- Actividad de Proyecto Formativo: Administrar la base de datos de acuerdo a los estándares y requisitos técnicos del cliente.
- Competencia: Administrar la base de datos de acuerdo a los estándares y requisitos técnicos del cliente.
- Resultados de Aprendizaje: Programar sentencias sql en un sistema manejador de bases de datos según requerimientos del cliente.

Duración de la Guía de Aprendizaje (horas): 74

1. PRESENTACIÓN

El **lenguaje SQL** es una herramienta esencial en el ámbito de las bases de datos relacionales, y su dominio abre un mundo de posibilidades más allá de las simples consultas. Con SQL avanzado, puedes:

- **Automatizar tareas:** Los **procedimientos almacenados** te permiten agrupar instrucciones SQL en bloques reutilizables, facilitando la gestión de tareas repetitivas.
- **Crear vistas personalizadas:** Estas son ideales para simplificar consultas complejas y proteger la estructura de la base de datos, brindando una capa de abstracción que mejora la seguridad y la eficiencia.



[tomado-de](#)

En esta guía, exploraremos temas como;

- Consultas multitable
- Vistas
- Procedimientos almacenados

Estas herramientas que enriquecerán las funcionalidades de tus bases de datos. ¡Aprovecha el potencial de SQL avanzado y lleva tus habilidades al siguiente nivel!

2. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

2.1 Actividades de Reflexión Inicial:

Descripción de la actividad:

Reconocer la importancia de las bases de datos relacionales en los sistemas de información. Lee atentamente, sigue las instrucciones y realiza la actividad correspondiente a este punto, que se encuentra en la carpeta *Material de apoyo*.

- ♦ **Material de apoyo:**
 - ♦ *31-TALL_ReflexionInicial*
- ♦ **Ambiente requerido:** Ambiente de formación.
- ♦ **Estrategias o técnicas didácticas activas:** Aprendizaje reflexivo y colaborativo.
- ♦ **Evidencia de aprendizaje:**
 - ♦ 31-RTA_ReflexionInicial_BaseDeDatosRelacional

2.2 Actividades de Contextualización e Identificación de Conocimientos:

Descripción de la actividad:

Entender los conceptos fundamentales de SQL y sus sentencias. Lee con atención, sigue las instrucciones y realiza la actividad correspondiente a este punto, que se encuentra en la carpeta *Material de apoyo*.

- ♦ **Material de apoyo:**

- ♦ 32-TALL_Contextualizacion
 - ♦ 32-LECT_SQL_TiposDeSentencias
 - ♦ 32-LECT_SQL_ConsultasMultitabla
 - ♦ 32-LECT_SQL_Vistas
 - ♦ 32-LECT_SQL_ProcedimientosAlmacenados

- ♦ **Ambiente de aprendizaje:** Ambiente de formación.

- ♦ **Estrategias didácticas:** Aprendizaje basado en la lectura y análisis, con enfoque creativo.

- ♦ **Evidencia de aprendizaje:**

- 32-RTA_Mandala_ConceptosClaveSQL
- 32-RTA_FrasesCreativas_TipoSentenciasSQL
- 32-RTA_UniendoSaberes_ConsultasMultitablaSQL
- 32-RTA_Metaforas_VistasSQL
- 32-RTA_CUarea_ProcedimientoAlmacenadoSQL

2.3 Actividades de Apropiación del Conocimiento:

Descripción de la actividad:

Realizar sentencias SQL para crear consultas multitable, vista y procedimientos almacenados en una base de datos. Lee con sumo detalle, sigue las instrucciones y realiza la actividad correspondiente a este punto, que se encuentra en la carpeta *Material de apoyo*.

- **Material de apoyo:**

- 33_TALL_Apropiacion

Ambiente de aprendizaje: Ambiente de formación.

Estrategias didácticas: Aprendizaje Activo y Visual – Enfoque Constructivista.

Evidencia de aprendizaje:

- 33-RTA_db_presupuesto_personal

- ModeloRelacional-db_presupuesto_personal.png
 - db_presupuesto_personal.sql

3.4 Actividades de Transferencia del Conocimiento

Descripción de la actividad:

Es posible que pienses que los temas que has aprendido son independientes entre sí, pero en realidad están conectados. En esta actividad, lo comprobarás. Lee con atención, sigue las instrucciones y realiza la actividad correspondiente a este punto, que se encuentra en la carpeta *Material de apoyo*.

- ♦ **Material de apoyo:**
 - 34_TALL_Transferencia
- ♦ **Ambiente de aprendizaje:** Ambiente de formación.
- ♦ **Estrategias didácticas:** Aprendizaje Basado en Problemas (ABP).
- ♦ **Evidencia de aprendizaje:**
 - 34-RTA_ScriptBaseDatos_Proyecto

4. EVIDENCIAS DE APRENDIZAJE

Entregables:

- ♦ 31-RTA_ReflexionInicial_BaseDeDatosRelacional
- ♦ 32-RTA_Mandala_ConceptosClaveSQL
- ♦ 32-RTA_FrasesCreativas_TipoSentenciasSQL
- ♦ 32-RTA_UniendoSaberes_ConsultasMultitablaSQL
- ♦ 32-RTA_Metaforas_VistasSQL
- ♦ 32-RTA_CUtarea_ProcedimientoAlmacenadoSQL
- ♦ 33-RTA_db_presupuesto_personal
 - ♦ ModeloRelacional-db_presupuesto_personal.png
 - ♦ db_presupuesto_personal.sql
- ♦ 34-RTA_ScriptBaseDatos_Proyecto

5. GLOSARIO DE TÉRMINOS

- ♦ **SQL:** Lenguaje de Consulta Estructurado, es un lenguaje estándar para acceder y manipular bases de datos.
- ♦ **DDL:** Lenguaje de Definición de Datos que define la estructura o esquema de la base de datos mediante comandos como `CREATE` , `DROP` , y `ALTER` .
- ♦ **DML:** Lenguaje de Manipulación de Datos que permite modificar los registros de la base de datos a través de operaciones como `INSERT` , `UPDATE` , y `DELETE` .
- ♦ **DQL:** Lenguaje de Consulta de Datos utilizado para recuperar datos de la base de datos, principalmente a través del comando `SELECT` .
- ♦ **DCL:** Lenguaje de Control de Datos utilizado para otorgar permisos en la base de datos, incluyendo comandos como `GRANT` y `REVOKE` .
- ♦ **TCL:** Lenguaje de Control de Transacciones que gestiona las transacciones dentro de la base de datos mediante comandos como `COMMIT` , `ROLLBACK` , y `SAVEPOINT` .
- ♦ **Consultas Multitabla:** Consultas que combinan datos de más de una tabla en una base de datos.
- ♦ **JOIN:** Comando utilizado para recuperar datos de dos o más tablas relacionadas mediante claves primarias y externas.
- ♦ **CROSS JOIN:** Tipo de JOIN que combina cada fila de una tabla con todas las filas de otra tabla, resultando en el producto cartesiano.
- ♦ **INNER JOIN:** Devuelve filas de ambas tablas que satisfacen una condición dada.
- ♦ **OUTER JOIN:** Incluye filas que no tienen coincidencias en la tabla unida, devolviendo `NULL` para esos registros.
- ♦ **LEFT JOIN:** Devuelve todas las filas de la tabla izquierda y las correspondientes de la tabla derecha, retornando `NULL` si no hay coincidencias.
- ♦ **RIGHT JOIN:** Devuelve todas las filas de la tabla derecha y las correspondientes de la tabla izquierda, retornando `NULL` si no hay coincidencias.
- ♦ **ON:** Cláusula utilizada para especificar la condición de unión entre dos tablas en un `JOIN` .
- ♦ **USING:** Cláusula similar a `ON` , pero requiere que los nombres de las columnas coincidentes sean idénticos en ambas tablas.
- ♦ **VISTAS:** Tablas virtuales que no almacenan datos propios, sino que muestran datos almacenados en otras tablas basadas en consultas SQL.
- ♦ **CREATE VIEW:** Instrucción SQL utilizada para crear una vista a partir de una consulta `SELECT`.
- ♦ **DROP VIEW:** Instrucción SQL utilizada para eliminar una vista existente.
- ♦ **CREATE OR REPLACE VIEW:** Instrucción SQL utilizada para actualizar una vista existente o crearla si no existe.
- ♦ **Procedimiento Almacenado:** Conjunto de instrucciones SQL almacenadas en la base de datos, que se pueden reutilizar y que pueden incluir parámetros de entrada y salida.
- ♦ **PARÁMETRO DE ENTRADA:** `INT` tipo de parámetro que envía datos al procedimiento almacenado.
- ♦ **PARÁMETRO DE SALIDA:** `OUT` tipo de parámetro que permite retornar datos al llamador una vez que el procedimiento ha terminado de ejecutarse.
- ♦ **PARÁMETRO DE ENTRADA Y SALIDA:** `INOUT` tipo de parámetro que puede ser utilizado tanto para enviar como para recibir datos en un procedimiento.

6. REFERENTES BIBLIOGRÁFICOS

- ♦ Brown, F. (2024, junio 28). *SQL commands: DML, DDL, DCL, TCL, DQL with query example*. Guru99. <https://www.guru99.com/sql-commands-dbms-query.html>
- ♦ Allen, M. (17 de julio de 2024). *MySQL JOINS tutorial: INNER, OUTER, LEFT, RIGHT, CROSS*. Guru99. <https://www.guru99.com/joins.html>
- ♦ W3Schools. (n.d.). *SQL CREATE VIEW Statement*. https://www.w3schools.com/sql/sql_view.asp
- ♦ Allen, M. (17 de julio de 2024). *MySQL Views: How to Create View from Tables with Examples*. Guru99. <https://www.guru99.com/views.html>
- ♦ W3Schools. (n.d.). *SQL stored procedures*. Recuperado de https://www.w3schools.com/mysql/sql_stored_procedures.asp
- ♦ MySQL Ya desde CERO. (n.d.). *Descripción de procedimientos almacenados en MySQL*. Recuperado de <https://www.tutorialesprogramacionya.com/mysqlya/temarios/descripcion.php?inicio=84&cod=101&punto=98>
- ♦ Software Testing Help. (n.d.). *MySQL stored procedure*. Recuperado de <https://www.softwaretestinghelp.com/mysql-stored-procedure/>

7. HISTORIAL DEL DOCUMENTO

Control del documento actual

- ♦ **Fecha de creación:** 2026-02-04
- ♦ **Autor:** Iván Oviedo
Cargo: Instructor
- ♦ **Modalidad:** Articulación con la Educación Media

Control de cambios del documento actual

(Completar solo si se hacen ajustes al documento)

- ♦ **Fecha de modificación:** 2026-02-04
- ♦ **Autor:** Iván Oviedo
- ♦ **Cargo:** Instructor
- ♦ **Modalidad:** Articulación con la Educación Media
- ♦ **Justificación del cambio:** Actualización de Guía por modificación de diseño curricular

Control del documento del formato GFPI-F135 V04 - GUÍA 04. TRANSACT

Fecha de creación: 2021-12-06

- ♦ **Autor:**
 - ♦ Giovanni Urrego
 - ♦ Gustavo Páramo
- ♦ **Cargo:** Instructor
- ♦ **Modalidad:** Articulación con la Educación Media

Control de cambios del formato GFPI-F135 V04 - GUÍA 04. TRANSACT

(Completar solo si se hacen ajustes al documento)

- ♦ **Fecha de modificación:** 2024-12-10
 - ♦ **Autor:**
 - Edwin Cañón
 - Fabian Hernandez
 - Henry Caicedo
 - Ramiro Romero
 - ♦ **Cargo:** Instructor
 - ♦ **Modalidad:** Articulación con la Educación Media
 - ♦ **Justificación del cambio:**
 - Migración al nuevo formato de guías y actualización del diseño curricular. Verificación material de apoyo y ajuste de actividades

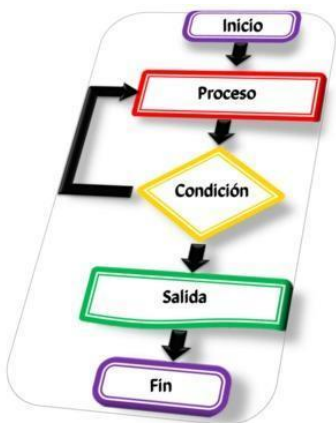


PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL
FORMATO GUÍA DE APRENDIZAJE
Guía 5 Lógica de programación y Algoritmia

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Técnico en Programación de Software.
- Código del Programa de Formación: 233104
- Nombre del Proyecto: desarrollo de aplicaciones de software para el sector empresarial
- Fase del Proyecto, Fase 3: Ejecución del Proyecto (Ejecución)
- Actividad de Proyecto: Actividad 5: construir interfaces y procesos lógicos de acuerdo con las necesidades del cliente.
- Competencias: *Técnicas* 220501096 - Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.
- Resultados de Aprendizaje Alcanzar: 2584211 - Resolver procesos lógicos a través de la implementación de algoritmos y el lenguaje de programación seleccionado.
- Duración de la Guía: 100 horas (80 horas presenciales y 10 horas desescolarizadas)

2. PRESENTACIÓN



SEQ Ilustración * ARABIC
1 [https://lucashiguitabedoya.ji
mdofree.com/2-periodo-](https://lucashiguitabedoya.jimdofree.com/2-periodo-)

Toda persona que se inicie en el mundo de la programación necesariamente debe tener en cuenta algunos elementos esenciales para llegar a ser un buen programador. La lógica es uno de estos elementos. Ésta es esencial en el conocimiento de la programación, porque es la base para escribir código que la computadora pueda interpretar correctamente. La lógica nos permite desarrollar aplicaciones utilizando una secuencia razonada de pasos para lograr determinado objetivo.

Esta guía le permitirá al aprendiz introducirse en el mundo de la programación a través de una serie de conocimientos que les darán las herramientas necesarias para obtener los procesos mentales adecuados para aprender a programar. Es de vital importancia que los aprendices realicen las actividades propuestas con conciencia, ya que esto les permitirá adquirir los conocimientos y la lógica necesaria de manera efectiva.



Los programadores no se hacen de la noche a la mañana, es un proceso que se puede ir realizando poco a poco y la lógica es uno de los aspectos que primero se deben desarrollar. Eso sí, hay que tener algunos conocimientos previos, por ejemplo, manejo

básico del computador y fundamentos de matemáticas, entre otros, también hay que tener mucha curiosidad y práctica constante.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial.

Actividad de Aprendizaje 05_1: Analizar los elementos básicos de la lógica proposicional y la construcción de algoritmos para entender el campo de aplicación de la lógica.

Tiempo estimado: 2 horas.

Descripción de la(s) actividad(es)

Antes de iniciar las actividades propuestas, conforme grupos, máximo de 2 personas y aliste una hoja de papel.

Observe del video “5.1¿Qué es PYTHON para qué SIRVE_.mp4” que se encuentra en la carpeta “material del aprendiz” o ingrese a la dirección <https://www.youtube.com/watch?v=lc5JJTQa4r8> y observe el video: que es Python y para qué sirve y tener un mejor concepto hacia dónde va orientada esta guía.

Detengamos el tiempo por un segundo y recordemos cómo se elabora una figura animal de papel. Sigamos los pasos que se les indican a continuación:

Parte 1:

Paso 1. Con la hoja de papel realice la figura de un animal (Mariposa, pavo real, elefante, pez, etc.).

Paso 2. Junto a sus compañeros de grupo escriba en inglés en una hoja como sería el paso a paso para elaborar la figura de un animal en papel.

Paso 3. ¡Ahora reflexionemos! ¿Qué pasaría si cambiamos los materiales o la forma del animal? ¿Cambiaría el paso a paso descrito? Escriban sus conclusiones utilizando el idioma inglés.

Parte 2:

¡Intentemos de nuevo! Con los mismos compañeros de grupo, analicen y escriban:

- a) ¿Cómo sería la solución paso a paso de la siguiente operación matemática? Dado el costo de un computador portátil y la cantidad de dinero entregado por el cliente, calcule e imprima el cambio que debe devolverse al cliente.
- b) ¿Cuáles conclusiones pueden obtener, relacionadas con el tema de la sesión de hoy, a partir de las actividades anteriores? Por favor, comenten en grupo y registren por escrito sus conclusiones.



Para finalizar la actividad en compañía del instructor, elijan a un compañero del grupo, que comparta la experiencia a los demás aprendices que se encuentran en el ambiente de formación (Parte 1 y Parte 2).

Parte 3:

- a) De forma individual van a ingresar al juego Bloxorz en el siguiente enlace <https://www.crazygames.com/es/game/bloxorz> y van a realizar una serie de pasos para ir avanzando de nivel
1. ¿En cuánto tiempo y serie de pasos realizo para avanzar al nivel 2?
 2. ¿Qué nivel fue complicado? ¿Qué nivel se desarrolló en menos tiempo?
- b) Instrucciones del juego de Nim:
1. Seleccione a dos personas para jugar.
 2. Dibujen en el tablero del salón de clases 12 óvalos formando un círculo.
 3. Por turnos, cada jugador deberá borrar uno o dos óvalos del círculo.
 4. Recuerden que solo se pueden borrar óvalos que estén juntos en el círculo y no se pueden saltar fichas.
 5. El objetivo del juego es ser el jugador que borre el último óvalo.
 6. Jueguen varias partidas e intercambien los roles para que todos tengan la oportunidad de jugar y analizar las estrategias.
 7. Observen y discutan juntos para identificar la estrategia ganadora, es decir, la mejor manera de jugar para asegurarse de ser quien borre el último óvalo.
 8. Diviértanse mientras ponen en práctica su lógica y estrategia en este desafiante juego de Nim en el tablero del salón de clases.

¡Que empiece la diversión y que el mejor estrategia gane en el tablero del salón de clases!

Referencia del juego de Nim: Sáenz de Cabezón, E. (11 de julio de 2018). Las matemáticas nos hacen más libres y menos manipulables [Video]. Aprendemos Juntos 2030. Recuperado el 10 de julio de 2023, de <https://youtu.be/BbA5dpS4Ccl?t=1587> (26:27).

Indicaciones para el entregable o evidencia: Crear un documento en formato PDF que contenga las respuestas a las preguntas de la Parte 1 y Parte 2 de las actividades de reflexión inicial. Guardar como **06_1_ActividadReflexiónInicial_LógicaDeProgramación_TuapellidoTunombre.pdf**. Recuerda seguir estas instrucciones para organizar y presentar adecuadamente tu entregable o evidencia.

Ambiente requerido: Aula de clase.

Materiales: Hojas de papel, lápices, esferos, tablero, marcadores, computadores, VideoBeam.

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



3.2.1 Actividad de Aprendizaje 06_2 Identificar los conocimientos necesarios para elaborar algoritmos básicos.

Tiempo estimado: 6 Horas.

Siga las indicaciones que se presentan a continuación:

Parte 1:

1. Haz clic en el siguiente enlace para acceder al juego "Voyage Aquatic – Code.org": <https://studio.code.org/s/aquatic/lessons/1/levels/1>. Sumérgete en esta aventura submarina y desarrolla tu lógica de programación mientras te diviertes.
2. **Objetivo:** Explora, resuelve desafíos y avanza en el juego utilizando la programación como herramienta principal.
3. **Dinámica:**
 - Juego individual: Supera los desafíos por tu cuenta y descubre nuevas formas de aplicar la lógica de programación en el entorno submarino.
 - Juego en parejas: Formen equipos y asignen roles. Uno controla el personaje y el otro analiza y escribe el código necesario.
4. **Comunicación y colaboración:**
 - ¡Trabajen juntos! Comuníquense, compartan ideas y tomen decisiones conjuntas para resolver los desafíos.
5. **Turnos de juego (solo para parejas):**
 - Alterna el control del personaje y la escritura de código para que ambos participen activamente.

¡Sumérgete en "Voyage Aquatic" y disfruta de esta experiencia desafiante y divertida mientras desarrollas tu lógica de programación!
6. A continuación, observe el siguiente video: “**05_VID_QuéEsUnAlgoritmo.mp4**” que se encuentra en el “**material de apoyo del aprendiz**”.
7. Comparte y socializa con tus compañeros si este último juego tiene relación con el video visto.

Parte 2:

¡Bienvenidos de vuelta! En actividades anteriores, hemos tenido la oportunidad de jugar juegos que otros han creado. Ahora, es el momento de cambiar de rol y poner en práctica nuestra creatividad y habilidades de programación por bloques para crear tu propio juego, un juego de laberinto.

1. ¡Accede a la plataforma MIT App Inventor en <https://appinventor.mit.edu/> y dale clic en la opción “Create Apps!” (¡Crear aplicaciones!).
2. Observa el video “**5_2COMOCREARUN LABERINTOEN MITAppInventor 2.mp4**” que se encuentra en la carpeta “**material del aprendiz**” para poder crear la aplicación app, sigue los pasos

Recuerda aprovechar los recursos y documentación disponibles en MIT App Inventor para ayudarte en el proceso de desarrollo.

¡Disfruta creando tu juego de laberinto en MIT App Inventor!



Indicaciones para el entregable o evidencia: Crear un documento en formato PDF que contenga las capturas de pantalla de los juegos de la Parte 1 y Parte 2. Guardar como **05_2_ActividadContextualización_LógicaDeProgramación_TuapellidoTunombre.pdf**. Recuerda seguir estas instrucciones para organizar y presentar adecuadamente tu entregable o evidencia.

3.2.2 Actividad de Aprendizaje 05_2 Identificar los conocimientos básicos necesarios para elaborar proposiciones simples y compuestas

Tiempo estimado: 4 Horas.

1. Observen el video “**05_VID_ProposicionesYOperacionesBásicas.mp4**” que se encuentra en el “**material de apoyo del aprendiz**”.
2. En grupos de dos personas realicen la lectura del documento denominado “**05_LECT_IntroducciónALaLógica.pdf**” que se encuentra en el “**material de apoyo del aprendiz**”.
3. Elaboren en su cuaderno 10 enunciados diferentes (en español y en inglés) y traten de observar si son o no proposiciones. Clasifíquelas en compuestas o simples al frente de cada enunciado.
4. Socialicen (en español y en inglés) con los demás grupos los enunciados trabajados y los resultados obtenidos.

3.2.3 Actividad de Aprendizaje: 06_2 Identificar los conocimientos básicos necesarios para manipular las entidades primitivas en programación como paso previo para iniciar el proceso de creación de algoritmos en Python

Tiempo estimado: 6 Horas.

La siguiente actividad se debe realizar en grupos de 2 aprendices.

1. Realicen la siguiente lectura “**05_LECT_EntidadesPrimitivas.pdf**” que se encuentra en el “**material de apoyo**”.
2. Observar el siguiente video “**05_VID_ResoluciónDeExpresiones_ (Lógica De La Programación).mp4**” se encuentra en el “**material de apoyo**”.
3. Descargar y resolver los ejercicios del taller “**51_TALL_ExpresionesMatemáticas.docx**” que se encuentra en el “**material de formación**”, aplicando los temas de la lectura y para el entregable o evidencia guardar como
51_TALL_ExpresionesMatemáticas_TuapellidoTunombre.pdf

3.2.4 Actividad de Aprendizaje 05_2 Identificar los conocimientos necesarios para elaborar algoritmos básicos.






Tiempo estimado: 7 Horas.

Realizar la siguiente actividad en español e inglés, deberá seguir las indicaciones que se presentan a continuación:

1. Conformar grupos de trabajo, máximo 3 personas.



2. Refresque su conocimiento sobre el tema observando nuevamente el video **“05_VID_QuéEsUnAlgoritmo.mp4”** que ya ha sido previamente visto.
3. También observe el siguiente video **“05_VID_LógicaDeProgramaciónEstructuraSelectiva.mp4”**.
4. A continuación, en cualquier aplicativo dibujen las figuras que se muestran a continuación:

Nombre: _____ 	Nombre: _____ 	 Nombre: _____	 Nombre: _____	 Nombre: _____
--	--	--	---	--

5. Luego, basándose en los videos vistos, identifiquen cada una de las figuras y describan su utilidad o funcionamiento. Indiquen en qué parte de un diagrama de flujo se pueden utilizar.
6. Realicen la lectura del siguiente material de apoyo que se encuentra en la carpeta material del aprendiz: **5 Instrucciones Básicas en un Algoritmo en python.pdf**.
7. A continuación, con el instructor desarrollara los dos ejercicios propuestos en Python en el archivo denominado **5_1Instrucciones_Basicas_Algoritmo_python.pdf** que se encuentra en la carpeta material del aprendiz

Ejemplo:

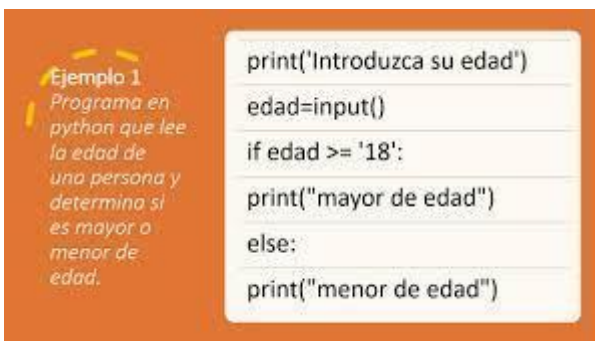


Ilustración 2: Diagrama de flujo tomado de
https://www.google.com/search?q=Instrucciones+Basicas+de+un+Algoritmo+en+python&sca_esv=d7e7d90bd6b3a957&udm=2&biw=1358&bih=608&ei=7aNZZ8vBBfSWwbkP8pOAwAc&ved=0ahUKEwiL7MHs-J-KAxV0SzABHfIJAHgQ4dUDCBE&uact=5&oq=Instrucciones+Basicas+de+un+Algoritmo+en+python&gs_l=EqNpbWciMEIuc3RyY3VjY2IvbmVzIEJhc2IjYXMgZGUgdW4gQWxnb3JpdG1vIGVulHB5dGhvbKivLFDgBVj_J3ABeACQAQCYAcoGoAG1F6oBCzAuMy41LjEuNi0xuAEDyAEA-AEBmAIAoAIAmAMA4gMFEgExIECIBgGSBwCgB8ID&sclient=img#vhid=pF7mdcXqTvrIOM&vssi d=mosaic

Hasta aquí analizamos problemas y cómo podemos darles la solución de diversas maneras. A continuación, nos enfocaremos en situaciones que involucran enunciados relacionados con la



programación, como el manejo de información sobre sueldos, personas, calificaciones, transacciones bancarias y más.

8. Visualizar el siguiente video **“5.EstructurasselectivasPython3.mp4”**, realizar el ejercicio en Python y luego en la carpeta material del aprendiz colocar en práctica desde python el ejercicio que se encuentra en el archivo denominado **“prueba_conocimiento.pdf”** que se encuentra en el **“material de formación”**.
9. Para el entregable o evidencia guardar en su portafolio de evidencia y aplicativo sena como **ejercicios_basicos_python.pdf**

Ambiente requerido: Aula de clase.

Materiales: Video beam, pantalla tv, cable HDMI, cable VGA, guía de formación, marcadores, resma de papel, tablero, conexión a internet, equipos de cómputo Visual Studio Code

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

3.3.1 Actividad de Aprendizaje 06_3 Construir tablas de verdad aplicando los conocimientos adquiridos para entender las respuestas lógicas obtenidas con el computador.

Tiempo estimado: 5 horas.

La actividad se debe realizar en grupos de 2 personas.

1. Observar el video **“05_VID_TablasDeVerdadProposicionesCompuestas.mp4”** contenido en la carpeta **“Material de Apoyo”**.
2. Leer el siguiente material de apoyo **“05_LECT_IntroducciónALaLógica.pdf”**.
3. En compañía del instructor realicen el taller que encontrarán en el documento denominado **“5_2_TALL_TablasDeVerdad.docx”** que se encuentra en el **“Material de Formación”**, el cual les permitirá aplicar los conocimientos adquiridos durante la sesión y para el entregable o evidencia guardar como **5_2_TALL_TablasDeVerdad_TuapellidoTunombre.pdf**.

3.3.2 Actividad de Aprendizaje 06_3 Emplear estructuras de programación secuenciales para la resolución de problemas

Tiempo estimado: 5 horas.

En grupos de máximo 3 personas realicen la siguiente actividad:

1. Observar el siguiente video **“05_VID_QuéEsUnAlgoritmo.mp4”** que se encuentra en el **“material de apoyo del aprendiz”**.
2. Leer el documento **“05_LECT_FundamentosYEstructurasSecuenciales.pdf”**, que se encuentra en el **“material de apoyo del aprendiz”**.
3. Revisar los ejemplos.
4. Realizar los ejercicios en el documento **“5_3_TALL_FundamentosYEstructurasSecuenciales_en python.docx”**, que se encuentra



en el “material de Formación” y para el entregable o evidencia guardar como **5_3_TALL_FundamentosYEstructurasSecuenciales_TuapellidoTunombre.docx**.

3.3.3 Actividad de Aprendizaje 06_3 Emplear estructuras de selección para crear algoritmos que puedan cambiar la secuencia de un proceso.

Tiempo estimado: 6 horas.

En grupos de máximo 3 personas realicen la siguiente actividad:

1. Observar el video “**05_VID_EstructurasCondicionalesEnAlgoritmos.mp4**”. Que se encuentra en el “material de apoyo del aprendiz”.
2. Leer el documento “**05_LECT_EstructurasDeSelección.pdf**”. con la explicación del instructor realizar los ejercicios que se encuentran en el documento.
3. Descargar y realizar el taller del archivo “**5_4_TALL_FundamentosYEstructurasSelectivas_python.docx**” que se encuentra en el “Material de Formación” y para el entregable o evidencia guardar como **64_TALL_FundamentosYEstructurasSelectivas_TuapellidoTunombre.pdf**.

3.3.4 Actividad de Aprendizaje 06_3 Emplear estructuras iterativas para crear algoritmos que puedan cambiar la secuencia de un proceso.

Tiempo estimado: 6 horas.

En grupos de máximo 2 personas realicen la siguiente actividad:

1. Leer el documento “**05_LECT_EstructurasDeRepetición.pdf**” Que se encuentra en el “material de apoyo del aprendiz”, revisar y analizar detenidamente los ejemplos que allí se encuentran.
2. Ingresar a la siguiente página y estudiar las lecciones 1, 2 y 3:
http://aprende.colombiaaprende.edu.co/sites/default/files/naspublic/curriculos_ex/n1g10_fproy/nivel1/programacion/unidad4/leccion1.html
3. Descargar y realizar el taller “**5_5_TALL_FundamentosYEstructurasIterativas.docx**” que se encuentra en la carpeta “material de formación” y para el entregable o evidencia guardar como **5_5_TALL_FundamentosYEstructurasIterativas_TuapellidoTunombre.pdf**.

3.3.5 Actividad de Aprendizaje 06_3 Cuestionario Lógica de Programación y Algoritmia para validar los conocimientos adquiridos por los aprendices.

Tiempo estimado: 1 hora.



En esta actividad se debe resolver el cuestionario de la Guía 5 que se encuentra en Zajuna. Para ello realice los siguientes pasos:

1. Ingrese a Zajuna.
2. Haga clic en el elemento con el icono del SENA y de nombre Programación de software.
3. En el panel de la izquierda haga clic en la opción Contenido del curso.
4. Haga clic en la carpeta actividades del proyecto. Y a continuación, haga clic en fase 3_Ejecucion.
5. luego haga clic en la carpeta competencia.
6. Haga clic en la Carpeta Guía 5 - Lógica de programación y algoritmia.
7. En esta carpeta encontrará un documento con el nombre de Cuestionario lógica programación Guía 5, haga clic en él. Y para empezar a desarrollar el cuestionario haga clic en el botón Continuar.
8. Al terminar haga clic en el botón enviar.

Ambiente requerido: Aula de informática.

Materiales: Video beam, pantalla tv, cable HDMI, cable VGA, guía de formación, marcadores, resma de papel, tablero, conexión a internet, equipos de cómputo, Visual Studio Code

3.4 Actividades de transferencia del conocimiento

Actividad de Aprendizaje 06_4: Aplicar los conceptos de Lógica de Programación y algoritmia en el proyecto seleccionado.

Tiempo estimado: 32 horas (25 presenciales, 7 desescolarizadas)

Por grupos de proyecto diseñe los diagramas de flujo y el pseudocódigo del módulo Usuarios en python que tiene el proyecto asignado manejando el registro, consulta y actualización de los datos ingresados.

Para el registro de un usuario el algoritmo debe pedir los datos del usuario a registrar, por ejemplo, nombre de usuario y contraseña y los demás datos necesarios, según lo indicado en el archivo plantilla del proyecto asignado.

Para la consulta de usuarios el algoritmo debe permitir consultar los usuarios según su número de documento de identificación.

Igualmente, para la actualización de los datos de un usuario, el algoritmo debe pedir el número de identificación del usuario al cual se le actualizarán los datos y luego se debe pedir el dato a actualizar.

Para la entrega de la evidencia, les solicitamos que coloquen los diagramas de flujo y pseudocódigos de su proyecto en un documento de Word y lo guarden como **05_InformeAplicaciónAlgoritmosProyecto_TuapellidoTunombre.pdf**.

Ambiente requerido: Aula de informática.

Materiales: Video beam, pantalla tv, cable HDMI, cable VGA, guía de formación, marcadores, tablero, conexión a internet, equipos de cómputo, Visual Studio Code.



4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
ejecución	construir interfaces y procesos lógicos de acuerdo con las necesidades del cliente	Construir algoritmos que conlleven a la solución de una situación planteada.	Evidencias de Conocimiento: <ul style="list-style-type: none"> 05_3 Cuestionario Lógica de Programación y Algoritmia 	Da respuesta sobre tablas de verdad, conectores lógicos, entidades primitivas, algoritmos, fases de los algoritmos, estructuras y su aplicación en la solución de un problema	Prueba de conocimientos cuestionario Zajuna
ejecución	construir interfaces y procesos lógicos de acuerdo con las necesidades del cliente	Construir algoritmos que conlleven a la solución de una situación planteada.	Evidencias de Desempeño: <ul style="list-style-type: none"> 5_1TALL_ExpresionesMatemáticas.pdf 52_TALL_TablasDeVerdad.pdf 53_TALL_FundamentosY EstructurasSecuenciales.pdf 44_TALL_FundamentosY Est 	<ul style="list-style-type: none"> Elabora Tablas de verdad de acuerdo con los problemas planteados Representa procesos del sistema a partir de la construcción de 	Observación Sistemática / Lista de Chequeo.



			<p>ructurasSelectivas.pdf</p> <p>55_TALL_FundamentosYEst ructurasIterativas.pdf</p>	<p>algoritmos, como parte de la solución a situaciones planteadas , utilizando lenguajes de programación orientados a objetos.</p> <p>Realizar algoritmos utilizando variables, constantes, variables, bucles, contadores, acumuladores, selección, seudocódigo.</p>	
ejecución	<p>construir interfaces y procesos lógicos de acuerdo con las necesidades del cliente</p>	<p>Construir algoritmos que conlleven a la solución de una situación planteada.</p>	<p>Evidencias de Producto:</p> <p>05_InformeAplicaciónAlgoritmosProyecto.docx</p>	<ul style="list-style-type: none"> • 	<p>Valoración de Producto Rubrica.</p>



5. GLOSARIO DE TÉRMINOS

Lectura: Consiste en recibir desde un dispositivo de entrada (p.ej. el teclado) un valor.

Escritura: Consiste en mandar por un dispositivo de salida (p.ej. monitor o impresora) un resultado o mensaje.

Programación: Es la implementación de un algoritmo en un determinado lenguaje de programación, para realizar un programa.

Algoritmo: Secuencia de pasos lógicos para dar solución a un problema, con un inicio y fin definidos.

Pseudocódigo: Es una descripción de alto nivel de un algoritmo que emplea una mezcla de lenguaje natural con algunas convenciones sintácticas propias de lenguajes de programación, como asignaciones, ciclos y condicionales.

Programa: (Software en inglés) es una secuencia de instrucciones que una computadora puede interpretar y ejecutar.

Variable: Espacio reservado de memoria cuyo valor puede cambiar durante la ejecución del programa.

Constante: Espacio reservado de memoria que permanece constante durante la ejecución del programa.

Diagrama de Flujo de Datos: Representación gráfica de un algoritmo.

Asignar: Establecer un valor a una variable.

Condicional: Estructura algorítmica de decisión que permite seguir un camino verdadero o falso en base a la o las condiciones.

Ciclo Anidado: Estructura algorítmica que consiste en ubicar un ciclo dentro de otro para realizar alguna tarea específica.

Índex: Variable de tipo entera encargada de controlar (contar) el número de iteraciones o veces que se repite un ciclo.



Ciclo: Estructura algorítmica que permite repetir una o varias líneas de código según se cumpla una condición.

Valor de verdad: Unidades para validación de los procesos de la lógica, puede ser Verdadero o Falso ó en su defecto Si o No ó 1 o 0.

6. REFERENTES BIBLIOGRÁFICOS

- Colombia aprende (sf). Currículos exploratorios en TIC. http://aprende.colombiaaprende.edu.co/sites/default/files/naspublic/curriculos_ex/n1g10_fproy/nivel1/programacion/unidad4/leccion1.html
- Instituto tecnológico de Sonora (2016). Pseudocódigo y PSEINT. https://www.itson.mx/oferta/isw/Documents/guia_pseint_2016.pdf
- Lógica matemática. Unidad 1. https://virtual.uptc.edu.co/ova/cursos/logica/unidad_uno/logica_unidad1.pdf

Prieto, S. (sf). Principios de algoritmia. <https://docplayer.es/42338535-Principios-de-algoritmia.html>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Jannez Urrego	Instructor	Articulación con la media	13/12/2021
Autor (es)	Maria Angelica Triviño	Instructor	Articulación con la media	02 de febrero 2017
Autor (es)	Giovanni Urrego Herrera	Instructor	Articulación con la media	15 de febrero de 2019
Autor (es)	Wilmer Cardozo Lopez Jennifer Fajardo	Instructor	Articulación con la media	08 de octubre de 2019



Autor (es)	Kevin García Aguilar	Instructor	Articulación con la media	05/12/2019
Autor (es)	Mauricio Lara r.	Instructor	Articulación con la media	30/11/2020
Autor (es)	Luis Felipe Martinez Guerrero	Instructor	Articulación con la media	13/12/2021

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Fabian Hernández	Instructor	Articulación con la media	03/07/2023	Actualización del diseño curricular.
Autor (es)	William Ramon Flórez	Instructor	Articulación con la media	11/12/2024	Actualización del diseño curricular.
Autor (es)	Joan Reyes Atencio	Instructor	Articulación con la media	11/12/2024	Actualización del diseño curricular.
Autor (es)	Bernardo Betancourt López	Instructor	Articulación con la media	12/02/2025	Actualización del diseño curricular.

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL
FORMATO GUÍA DE APRENDIZAJE
GUÍA 6. Diagrama de Clases Programación Orientada a Objetos POO Y Desarrollo Back –
End – Framework - Python

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Técnico en programación de software.
- Código del Programa de Formación: 233104 v1.
- Nombre del Proyecto: Desarrollo de aplicaciones de software para el sector empresarial.
- Fase del Proyecto: Fase 3 - Ejecución del Proyecto (Ejecución).
- Actividad de Proyecto: Actividad 5: Realizar CRUD validando la conexión con la Base de Datos en la arquitectura estipulada
- Competencia: 220501097 - Construir algoritmos que conlleven a la solución de una situación planteada.
- Duración de la Guía: Total 240 horas, trabajo directo 192 horas, 48 horas de trabajo independiente.

2. PRESENTACIÓN

2.1 El diagrama de clases UML

El diagrama de clases es uno de los elementos fundamentales del Lenguaje Unificado de Modelado (UML), ya que permite representar de forma gráfica la estructura estática de un sistema orientado a objetos. A través de este tipo de diagrama se describen las clases, sus atributos, métodos y las relaciones que existen entre ellas, facilitando la comprensión, el análisis y el diseño de soluciones de software antes de su implementación.

En el desarrollo de esta actividad, se espera que el aprendiz comprenda la importancia de los diagramas de clases como herramienta de modelado, comunicación y documentación dentro del proceso de desarrollo de software. Asimismo, deberá identificar cómo los conceptos de clases, objetos, herencia, asociación, agregación, composición y multiplicidad se reflejan de manera visual en UML, permitiendo representar sistemas reales de forma clara y estructurada.



Diagrama UML ilustrativo. (s. f.). [Imagen]. Recuperado de URL <https://media.istockphoto.com/id/492699028/es/vector/uml-lenguaje-unificado-de-modelado-de-trabajo-en-equipo-de-dise%C3%B1o-de-modelado-sistema-de.jpg>

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 ACTIVIDAD DE REFLEXIÓN INICIAL:

Toda persona, animal o cosa puede describirse a partir de sus características y comportamientos, por esto, consideremos una persona que viva con nosotros, nuestro mejor amigo o amiga y nuestra mascota, si no tiene mascota, considere el animal que le gustaría tener como mascota, ahora, por cada uno de ellos plantee las características principales que los describen y sus principales comportamientos, adicione mínimo 6 características y 4 comportamientos por cada uno de ellos. Esta actividad hace parte del entregable de la primera actividad como evidencia de la guía DE APRENDIZAJE 6.

3.1.1 Actividad de Aprendizaje 06_1: Reconocer los componentes del Diagrama de Clases según estándares UML.

Ahora, le invitamos a dar lectura al archivo **06_Introduccion_Diagrama_Clases** ubicado en: **MATERIAL DE APOYO** – Diagrama de clases y sus elementos, a partir de la lectura señalada, de manera individual, le invitamos a retomar el listado de características y comportamientos de la sección Reflexión Inicial, en el que seleccionó una persona que vive con usted, su mejor amigo o amiga y su mascota, cree las clases necesarias que permitan crear objetos tipo familiar, mejor amigo y mascota, recuerde, debe emplear la estructura adecuada presentada en el documento de lectura para definir clases.

Seguido a esto lo invitamos a “ver el **video diagrama-de-clases-uml---tutorial.mp4** ubicado en la carpeta **Material de apoyo**

Una vez visualizado el video el aprendiz tendrá que identificar las diferentes clases de la actividad inicial con las cuales creará el diagrama de clases lo cual le permitirá aplicar los conocimientos adquiridos durante la lectura y la visualización del video analizando cuáles son sus respectivas relaciones.

NOMBRE_CLASE
[visibilidad] nombre_atributo [: tipo_dato] [= valor_inicial]
[visibilidad] nombre_metodo [(lista_parametros)] [: valor_devuelto]

Esta actividad hará parte del entregable EVIDENCIA GUÍA DE APRENDIZAJE 6. Una vez terminada la actividad, el instructor dará las indicaciones para que sea socializada con todo el grupo de aprendices.

Guardar actividad como: Evidencia_Aprendizaje_Componentes_diagrama_de_clase.pdf

Ambiente Requerido: Ambiente de formación con equipos de cómputo y acceso a internet.

Materiales y/o Recursos: Libreta de apuntes por aprendiz.

Tiempo para la actividad: 2 horas.

3.1.2 Actividad de Aprendizaje 06_2: Diseñar el Diagrama de Clases partiendo del conocimiento previo.

Antes de continuar, con la información comprendida del video **diagrama-de-clases-uml---tutorial** ubicado en la carpeta **Material de apoyo**

Luego debe dar lectura al archivo **06_Elementos_Diagrama_Clases**. Ubicada en la carpeta **material de apoyo**, a partir de la lectura señalada y la información del video previamente visto, diseñe el diagrama de clases de su proyecto identificando las diferentes clases y sus relaciones. Esta actividad hace parte del entregable EVIDENCIA GUÍA DE APRENDIZAJE 6. Una vez terminada la actividad, el instructor dará las indicaciones para que sea socializada con todo el grupo de aprendices.

Guardar actividad como: Evidencia_Aprendizaje_Diagrama_de_Clase_Proyecto.pdf

Ambiente Requerido: Ambiente de formación con equipos de cómputo y acceso a internet.

Materiales y/o Recursos: Libreta de apuntes por aprendiz. Día, lucidchart, Canva.

Tiempo para la actividad: 2 hora.

3.3. ACTIVIDADES DE CONTEXTUALIZACIÓN E IDENTIFICACIÓN DE CONOCIMIENTOS NECESARIOS PARA EL APRENDIZAJE.

La Programación Orientada a Objetos (POO) es un paradigma que organiza el código en torno a objetos, los cuales combinan datos y comportamientos. Explica con tus propias palabras los cuatro pilares fundamentales de la POO: encapsulamiento, herencia, polimorfismo y abstracción. Luego, crea un ejemplo práctico en el lenguaje de programación de tu preferencia donde se evidencie el uso de por lo menos dos de estos pilares. Finalmente, describe cómo tu ejemplo refleja los conceptos teóricos desarrollados en la primera parte de la actividad.

Pero antes de continuar, se debe tener presente que esta guía se basa en la Programación Orientada a Objetos (POO), estilo de programación que se basa en el concepto de clases y objetos.

Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables que serán los planos base de código y que llamaremos clases, las cuales permitirán crear instancias individuales que llamaremos objetos.

DESCRIPCION DE LAS ACTIVIDADES:

3.2.1. CLASES Y OBJETOS (POO)

En la Programación Orientada a Objetos, un objeto se construye a partir de características (atributos) y comportamientos (métodos).

Antes de continuar, ver el video “**Introducción a la Programación Orientada a Objetos.mp4**” y realice la lectura **06_Conceptos_Básicos_POO** ubicados ambos en la carpeta **Material de apoyo**.

Con base a la información del video para comprender este concepto desde un contexto cotidiano, piense en tres elementos cercanos a usted:

- Un objeto físico de uso diario (por ejemplo: un celular, una bicicleta, una mochila, etc.)
- Un lugar (por ejemplo: su casa, el salón de clase, un parque, etc.)
- Un servicio digital que utilice con frecuencia (por ejemplo: WhatsApp, Netflix, Google Maps, etc.)

Ahora, para cada uno de los tres elementos, identifique y describa:

- Mínimo 6 características (atributos) que permiten representarlo.
- Mínimo 4 comportamientos (métodos) que dicho objeto podría realizar o permitir.

Una vez terminada la actividad, el instructor dará las indicaciones necesarias para que sea socializada la información y comprendida con todo el grupo de aprendices.

Esta actividad hace parte del entregable **EVIDENCIA GUÍA DE APRENDIZAJE 6.**

Guardar actividad como:

6_3_POO_Clases_y_Objeto.pdf en su carpeta evidencias / **Guía 6.**

Ambiente Requerido: Ambiente de formación con equipos de cómputo y acceso a internet.

Materiales y/o Recursos: Libreta de apuntes por aprendiz. Visual Studio Code

Tiempo para la actividad: 4 horas.

3.2.2 Actividad de Aprendizaje 06_1: CLASES Y OBJETOS (POO):

Antes de iniciar el desarrollo práctico de clases y objetos en Programación Orientada a Objetos (POO), es fundamental que el aprendiz comprenda los conceptos teóricos y la estructura básica que conforma una clase. Así como los diferentes pilares de la POO herencia, polimorfismo.

Para comprender este concepto desde la práctica, realice lo siguiente:

Para esta actividad, le invitamos a realizar la lectura del material de apoyo relacionado con Programación Orientada a Objetos (POO), específicamente el documento referente a la **Estructura y definición de clases, atributos y métodos.pdf**, ubicado en la sección *Material de apoyo*. A partir de la lectura señalada, y de manera individual realice los talleres que se encuentran en la carpeta *Material de apoyo*:

Actividad:

- **Taller_06_4_Clases_Objeto**
- **Taller_06_6_Polimorfismo**
- **Taller_06_5_Herencia**

Esta actividad hace parte del entregable EVIDENCIA GUÍA DE APRENDIZAJE 6.

Guarde la actividad como:

Comprima las carpetas de las actividades desarrolladas y guárdela en una sola carpeta principal con el nombre **Pilares_de_la_POO** y guárdela en su carpeta evidencias / Guía 6.

3.3 ACTIVIDADES DE APROPIACIÓN:

3.3.1 Actividad de Aprendizaje 06_8 FRAMEWORKS Y ENTORNOS VIRTUALES

Framework

Un Framework es un conjunto de herramientas, librerías y buenas prácticas que facilitan el desarrollo de software. En lugar de programar todo desde cero, un Framework ofrece una estructura ya definida sobre la cual el programador puede construir su aplicación, siguiendo reglas

y patrones establecidos. Esto permite ahorrar tiempo, mejorar la organización del código y garantizar que se usen métodos más seguros y eficientes.

Actividad: “Estructura de una aplicación en Flask y Rutas en python”

Para comprender estos conceptos desde la práctica, realice lo siguiente:

Para esta actividad, le invitamos a consultar el **Material de apoyo/Tutoriales Flask**

Los videos

- **Tutorial Flask parte 1**

A partir de la información comprendida en los video tutoriales, y de manera individual, realice la construcción de la estructura de una aplicación, la cual deberá realizar siguiendo el paso a paso explicado en los video tutoriales y de esta manera comprender el trabajo con Frameworks (Flask), entornos virtuales y la creación de rutas.

Esta actividad hace parte del entregable EVIDENCIA GUÍA DE APRENDIZAJE 6.

Guarde la actividad de la siguiente manera:

Comprima la carpeta de la aplicación realizada y guárdela con el nombre **Estructura_aplicacion_flask_rutas_python** en su carpeta evidencias de la guía.

3.3.2 Actividad de Aprendizaje emplear los conceptos relacionados de Frameworks aplicados al desarrollo de aplicaciones web en PYTHON

Descripción de las actividades.

Actividad: “Aplicación en Flask y Python”

- Continuando con la estructura realizada con base al tutorial anterior, sigue desarrollando la aplicación web para comprender las ventajas de usar Frameworks y rutas en el desarrollo de aplicaciones mediante python, realice la actividad: “**Tutorial Flask parte 2**”,

Esta actividad hace parte del entregable EVIDENCIA GUÍA DE APRENDIZAJE 6.

Guarde la actividad de la siguiente manera:

Comprima la carpeta de la aplicación realizada y guárdela con el nombre **Aplicacion_flask_python** en su carpeta evidencias / Guía 6.

Ambiente Requerido: Ambiente de formación con equipos de cómputo y acceso a internet.

Materiales y/o Recursos: computador, Xampp, visual studio code, Flask, Python.

Tiempo para la actividad: 10 hora.

3.3.3 Actividad de Aprendizaje desarrollo de aplicaciones web en PYTHON con POO y conexión a la base de datos MySQL.

- Realice paso a paso la actividad del videotutorial “**Taller_Proyecto_conexión_base_de_datos**”, implementando rutas y conexión a base de datos utilizando la **POO**.

Ambiente requerido: Ambiente de formación.

Guarde la actividad de la siguiente manera:

Comprima y guarde la carpeta del archivo ejecutable del proyecto con el nombre “**Flask_conexion_base_de_datos.**” y súbalo al portafolio del aprendiz en la carpeta asignada carpeta evidencias / Guía 6.

Ambiente Requerido: Ambiente de formación con equipos de cómputo y acceso a internet.

Materiales y/o Recursos: computador, Xampp, visual studio code, Flask, Python.

Tiempo para la actividad: 10 hora.

3.3.4 Actividad de Aprendizaje Implementación de un Login Usando Flask, Python y MySQL

Actividades de apropiación del conocimiento:

De acuerdo con el material compartido y las prácticas de desarrollo con el Framework Flask, el aprendiz deberá implementar un módulo de autenticación (Login) utilizando Python y MySQL.

- Basado en el material de apoyo disponible (videos, documentos y páginas web) y el ejemplo de implementación de autenticación en Flask, el aprendiz deberá desarrollar un sistema de inicio de sesión para una aplicación web.
- Diseñar los formularios básicos de registro de usuario e inicio de sesión utilizando HTML, CSS y Java Script, incluyendo validación de datos.
- Crear la estructura de base de datos para almacenar usuarios y desarrollar la clase de conexión a MySQL.
- Implementar la validación de credenciales y control de acceso a páginas protegidas.
- Esta actividad es de carácter grupal según la conformación del proyecto y todo el código desarrollado debe estar organizado mediante métodos y clases, además de estar debidamente documentado.

Para esta actividad, le invitamos a realizar la consulta del material de apoyo el video tutorial con el nombre **TUTORIAL_LOGIN Primera parte. mp4** y **TUTORIAL_LOGIN Segunda parte. mp4** relacionado con Frameworks (Flask) entornos virtuales y rutas, A partir de los video tutoriales como guía y de manera grupal, realice la actividad.

Comprima y guarde la carpeta del archivo ejecutable del proyecto con el nombre **"Flask_Login"** y súbalo al portafolio del aprendiz en la carpeta asignada carpeta evidencias / Guía 6.

Ambiente Requerido: Ambiente de formación con equipos de cómputo y acceso a internet.

Materiales y/o Recursos: computador, Xampp, visual studio code, Flask, Python.

Tiempo para la actividad: 4 hora.

3.4 Actividad Transferencia de Conocimiento. Creando una Aplicación Web Desde Cero Usando el Framework Flask, Python y MySQL

Descripción de la Actividad

Actividades de Apropiación del conocimiento de acuerdo al documento compartido y a las prácticas de construcción de la arquitectura de acuerdo con el FRAMEWORK FLASK el aprendiz deberá realizar la estructura CRUD utilizando el lenguaje de programación Python y Mysql.

- Basado en el material de apoyo disponible (Videos, documentos y páginas web) y el ejemplo "Creando una Aplicación Web Desde Cero Usando Python Flask y MySQL.
- Se tendrán que diseñar los formularios básicos para la CRUD del proyecto asignado.
- Posteriormente se debe crear la clase de conexión a base de datos usando Mysql.
- Esta actividad es de carácter grupal según como esté conformado el proyecto y se debe tener en cuenta que todo el código programado debe estar con (métodos, clases, atributos) y adicionalmente debe estar documentado en su totalidad.

Ambiente requerido: Ambiente de formación

Materiales: computador, Xampp, MicroSoft visual studio code,

Tiempo Estimado: 20 horas.

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
<ul style="list-style-type: none"> • Evidencias de Conocimiento: Evidencia_Aprendizaje_Componentes_diagrama_de_clase <p>Evidencias de Desempeño:</p> <ul style="list-style-type: none"> • Evidencia de Aprendizaje_POO_Clases_y_Objetos_Pilares_de_la_POO <p>Evidencias de Producto:</p> <ul style="list-style-type: none"> • Estructura_aplicacion_flask_rutas_python • Estructura_aplicacion_flask_rutas_python • Aplicacion_flask_python • Flask_conexion_base_de_datos • Flask_Login 	<p>Comprende los conceptos básicos de la programación orientada a objetos: el alumno demuestra entender las ventajas y desventajas de este paradigma, así como sus conceptos centrales como abstracción, encapsulamiento, herencia y polimorfismo.</p> <p>Comprende la teórica de la arquitectura de un framework: el alumno identifica correctamente los principios y funcionamiento del patrón Modelo-Vista-Controlador en el desarrollo de aplicaciones web.</p> <p>Aplicación de POO y bases de datos: el alumno aplica adecuadamente los principios de la programación orientada a objetos, así como las técnicas de conexión a bases de datos mediante PDO en el desarrollo de software.</p> <p>Reconoce el valor de los patrones de diseño: el alumno compara aplicaciones con y sin patrones de diseño, reconociendo las ventajas en mantenibilidad, escalabilidad y buenas prácticas que proporcionan.</p> <p>Desarrolla aplicaciones web con entornos virtuales</p>	<p>Formulación de Preguntas</p> <p>Instrumento Cuestionario</p>

	El alumno implementa correctamente la estructura de un framework en el desarrollo de aplicaciones web funcionando de forma adecuada.	
--	--	--

5. GLOSARIO DE TÉRMINOS

Clase: modelo o plano para la creación de objetos que comparten atributos y métodos. Define las características y el comportamiento.

Objeto: instancia individual de una clase, con valores concretos para sus atributos.

Atributo: variable que forma parte de la definición de la clase, caracteriza al objeto.

Método: función o procedimiento asociado a una clase, define su comportamiento.

Abstracción: proceso mental de aislar un elemento de su contexto y considerarlo de forma independiente.

Encapsulamiento: empaquetar atributos y métodos dentro de una clase. Ocultación de detalles de implementación.

Herencia: mecanismo que permite basar una clase en otra ya existente, heredando sus atributos y métodos.

Polimorfismo: propiedad por la que objetos de diferentes clases pueden responder a un mismo mensaje cada uno a su manera.

Modelo: representa los datos, lógica de negocio y reglas de la aplicación.

Vista: interfaz de usuario. Presenta el modelo y captura la entrada del usuario.

Controlador: intermediario entre el modelo y las vistas. Gestiona la lógica y el flujo de datos.

Enrutador: gestiona las peticiones entrantes y las redirecciona al controlador apropiado.

Patrón: solución reutilizable a un problema común en el desarrollo de software.

Instancia: objeto concreto creado a partir de una clase. Tiene estado y comportamiento.

Interfaz: conjunto de métodos y propiedades que caracterizan un tipo de objeto.

Modularidad: principio de diseño, divide un sistema en partes separadas y discretas (módulos)

Encapsulamiento: empaquetar y ocultar partes independientes de un sistema.

Acoplamiento: grado de interdependencia entre módulos de un sistema.

Cohesión: grado en que partes de un módulo están conectadas.

Reutilización: uso de código existente en nuevas aplicaciones.

Herencia múltiple: clase que hereda de más de una clase padre.

6. REFERENTES BIBLIOGRÁFICOS

- El paradigma de Programación Orientada a Objetos en ASP y el patrón de arquitectura MVC:
- Documentación oficial PYTHON.. Disponible en:
- <https://aws.amazon.com/es/what-is/python/>
- Fundamentos básicos PYTHONDisponible en:
- <https://softwarecrafters.io/python/tutorial-de-python-introduccion>
- Sintaxis básica PDO. 30 de Enero de 2018. Disponible en:

<https://diego.com.es/tutorial-de-pdo>

- Manual PYTHON. Disponible en:
- <https://persoal.citius.usc.es/eva.cernadas/informaticaparacientificos/material/libros/Python%20para%20todos.pdf>

Videos

PYTHON DESDE CERO:

https://www.educacionit.com/cursos-de-python?gad_source=2&gclid=EAlalQobChMIrf6kk8zNiwMVsEtHAR3F4iBpEAAYASAAEgKWGfD_BwE&utm_campaign=co_conv_search_pyd_python_ca_12024___&utm_medium=search&utm_source=google

CLASES Y OBJETOS EN PYTHON

<https://www.youtube.com/watch?v=nKPbfIU442g>

CLASES Y OBJETOS EN PYTHON

<https://www.youtube.com/watch?v=9x7RK6mb1uA>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Jannez Milson Urrego Quiroz.	Instructor	Coordinación	13/12/2021
	Pablo Suarez			
	Oscar Jamir Ortega			
	Bernardo Betancourt			

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Magaly Iozano Beltrán	Instructor	Articulación	01/12/2024	Actualización Guía
	Edwin Cañon	Instructor	Articulación	01/12/2024	Actualización actividad 3.3
	Jannez Urrego	Instructor	Articulación	24/11/2025	Actualización actividad 6_3_2_4_LOGIN
	Pablo Suarez	Instructor	Articulación	13/12/2023	Actualización Guía

GUIA DE APRENDIZAJE	EVIDENCIAS	FÍSICO	DIGITAL	FECHA DE ENTREGA
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_1_ActividadReflexiónInicial_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2_ActividadContextualización_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para elaborar proposiciones simples y compuestas		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para manipular las entidades primitivas en programación como paso previo para iniciar el proceso de creación de algoritmos en Python		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos necesarios para elaborar algoritmos básicos.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Construir tablas de verdad aplicando los conocimientos adquiridos para entender las respuestas lógicas obtenidas con el computador.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de programación secuenciales para la resolución de problemas		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de selección para crear algoritmos que puedan cambiar la secuencia de un proceso.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Cuestionario Lógica de Programación y Algoritmia para validar los conocimientos adquiridos por los aprendices.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_4: Aplicar los conceptos de Lógica de Programación y algoritmia en el proyecto seleccionado.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	2.1 El diagrama de clases UML y la Programación Orientada a Objetos: PRESENTACION		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1. ACTIVIDAD DE REFLEXIÓN INICIAL		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.1 Actividad de Aprendizaje 6_1: Interpretar el Diagrama de Clases partiendo del conocimiento previo.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.2 Actividad de Aprendizaje 01_2: Reconocer los componentes del Diagrama de Clases según estándares UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.3 Actividad de Aprendizaje 01_3 Adicionar relaciones al Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.4 Actividad de aprendizaje 01_6 Identificar las clases y restricciones de un Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.5 Actividad de Ejercicios iniciales sobre conceptos de Programación básica en php		X	



GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.6 Analizar la importancia de la programación orientada a objetos y la importancia de los requisitos para la aplicación.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.7 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.8 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.1 Actividad de Aprendizaje 06_1 Analizar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.2 Actividad de Aprendizaje 06_2 Interpretar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.3 Asociar los conceptos relacionados con el patrón de diseño MVC aplicados al desarrollo de aplicaciones web en PYTHON		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.4 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.5 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.6 Realizar el siguiente ejemplo ubicado en el “taller 06_2_Programacion_Orientada_objetos”		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.7 Realizar el siguiente ejemplo ubicado en el taller 06_03_Programacion_Orientada_objetos2		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.8 Realizar el siguiente ejemplo ubicado en el “7_1_LECTURA_Entendiendo_MVC_en_PYTHON”,		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.1 Actividad de Aprendizaje 03_1 Determinar los datos e información necesaria para el diseño del Diagrama de Clases		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.2 Actividad de Aprendizaje 06_2 Diseñar Diagramas de Clases para diferentes problemas reales según parámetros establecidos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.3. Actividades de apropiación del conocimiento (Conceptualización y Teorización). Creando una Aplicación Web Desde Cero Usando Python Flask y MySQL _ Envato Tuts+.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.4. Actividades de apropiación del conocimiento: Comprender la implementación del patrón Modelo Vista controlador en aplicaciones web Php python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.5 Actividades de apropiación del conocimiento: Implementar el patrón Modelo Vista controlador en aplicaciones web Python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.4.1 Actividad de Aprendizaje 04_1 Definir las clases y elementos fundamentales para el diseño del proyecto		X	

[illegible]

GUIA DE APRENDIZAJE	EVIDENCIAS	FÍSICO	DIGITAL	FECHA DE ENTREGA
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_1_ActividadReflexiónInicial_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2_ActividadContextualización_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para elaborar proposiciones simples y compuestas		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para manipular las entidades primitivas en programación como paso previo para iniciar el proceso de creación de algoritmos en Python		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos necesarios para elaborar algoritmos básicos.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Construir tablas de verdad aplicando los conocimientos adquiridos para entender las respuestas lógicas obtenidas con el computador.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de programación secuenciales para la resolución de problemas		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de selección para crear algoritmos que puedan cambiar la secuencia de un proceso.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Cuestionario Lógica de Programación y Algoritmia para validar los conocimientos adquiridos por los aprendices.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_4: Aplicar los conceptos de Lógica de Programación y algoritmia en el proyecto seleccionado.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	2.1 El diagrama de clases UML y la Programación Orientada a Objetos: PRESENTACION		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1. ACTIVIDAD DE REFLEXIÓN INICIAL		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.1 Actividad de Aprendizaje 6_1: Interpretar el Diagrama de Clases partiendo del conocimiento previo.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.2 Actividad de Aprendizaje 01_2: Reconocer los componentes del Diagrama de Clases según estándares UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.3 Actividad de Aprendizaje 01_3 Adicionar relaciones al Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.4 Actividad de aprendizaje 01_6 Identificar las clases y restricciones de un Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.5 Actividad de Ejercicios iniciales sobre conceptos de Programación básica en php		X	



GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.6 Analizar la importancia de la programación orientada a objetos y la importancia de los requisitos para la aplicación.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.7 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.8 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.1 Actividad de Aprendizaje 06_1 Analizar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.2 Actividad de Aprendizaje 06_2 Interpretar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.3 Asociar los conceptos relacionados con el patrón de diseño MVC aplicados al desarrollo de aplicaciones web en PYTHON		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.4 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.5 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.6 Realizar el siguiente ejemplo ubicado en el “taller 06_2_Programacion_Orientada_objetos”		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.7 Realizar el siguiente ejemplo ubicado en el taller 06_03_Programacion_Orientada_objetos2		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.8 Realizar el siguiente ejemplo ubicado en el “7_1_Lectura_Entendiendo_MVC_en_PYTHON”,		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.1 Actividad de Aprendizaje 03_1 Determinar los datos e información necesaria para el diseño del Diagrama de Clases		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.2 Actividad de Aprendizaje 06_2 Diseñar Diagramas de Clases para diferentes problemas reales según parámetros establecidos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.3. Actividades de apropiación del conocimiento (Conceptualización y Teorización). Creando una Aplicación Web Desde Cero Usando Python Flask y MySQL _ Envato Tuts+.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.4. Actividades de apropiación del conocimiento: Comprender la implementación del patrón Modelo Vista controlador en aplicaciones web Php python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.5 Actividades de apropiación del conocimiento: Implementar el patrón Modelo Vista controlador en aplicaciones web Python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.4.1 Actividad de Aprendizaje 04_1 Definir las clases y elementos fundamentales para el diseño del proyecto		X	

[illegible]

GUIA DE APRENDIZAJE	EVIDENCIAS	FÍSICO	DIGITAL	FECHA DE ENTREGA
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_1_ActividadReflexiónInicial_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2_ActividadContextualización_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para elaborar proposiciones simples y compuestas		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para manipular las entidades primitivas en programación como paso previo para iniciar el proceso de creación de algoritmos en Python		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos necesarios para elaborar algoritmos básicos.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Construir tablas de verdad aplicando los conocimientos adquiridos para entender las respuestas lógicas obtenidas con el computador.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de programación secuenciales para la resolución de problemas		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de selección para crear algoritmos que puedan cambiar la secuencia de un proceso.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Cuestionario Lógica de Programación y Algoritmia para validar los conocimientos adquiridos por los aprendices.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_4: Aplicar los conceptos de Lógica de Programación y algoritmia en el proyecto seleccionado.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	2.1 El diagrama de clases UML y la Programación Orientada a Objetos: PRESENTACION		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1. ACTIVIDAD DE REFLEXIÓN INICIAL		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.1 Actividad de Aprendizaje 6_1: Interpretar el Diagrama de Clases partiendo del conocimiento previo.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.2 Actividad de Aprendizaje 01_2: Reconocer los componentes del Diagrama de Clases según estándares UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.3 Actividad de Aprendizaje 01_3 Adicionar relaciones al Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.4 Actividad de aprendizaje 01_6 Identificar las clases y restricciones de un Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.5 Actividad de Ejercicios iniciales sobre conceptos de Programación básica en php		X	



GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.6 Analizar la importancia de la programación orientada a objetos y la importancia de los requisitos para la aplicación.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.7 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.8 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.1 Actividad de Aprendizaje 06_1 Analizar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.2 Actividad de Aprendizaje 06_2 Interpretar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.3 Asociar los conceptos relacionados con el patrón de diseño MVC aplicados al desarrollo de aplicaciones web en PYTHON		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.4 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.5 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.6 Realizar el siguiente ejemplo ubicado en el “taller 06_2_Programacion_Orientada_objetos”		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.7 Realizar el siguiente ejemplo ubicado en el taller 06_03_Programacion_Orientada_objetos2		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.8 Realizar el siguiente ejemplo ubicado en el “7_1_Lectura_Entendiendo_MVC_en_PYTHON”,		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.1 Actividad de Aprendizaje 03_1 Determinar los datos e información necesaria para el diseño del Diagrama de Clases		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.2 Actividad de Aprendizaje 06_2 Diseñar Diagramas de Clases para diferentes problemas reales según parámetros establecidos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.3. Actividades de apropiación del conocimiento (Conceptualización y Teorización). Creando una Aplicación Web Desde Cero Usando Python Flask y MySQL _ Envato Tuts+.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.4. Actividades de apropiación del conocimiento: Comprender la implementación del patrón Modelo Vista controlador en aplicaciones web Php python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.5 Actividades de apropiación del conocimiento: Implementar el patrón Modelo Vista controlador en aplicaciones web Python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.4.1 Actividad de Aprendizaje 04_1 Definir las clases y elementos fundamentales para el diseño del proyecto		X	

[illegible]

GUIA DE APRENDIZAJE	EVIDENCIAS	FÍSICO	DIGITAL	FECHA DE ENTREGA
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_1_ActividadReflexiónInicial_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2_ActividadContextualización_LógicaDeProgramación_TuapellidoTunombre.pdf		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para elaborar proposiciones simples y compuestas		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos básicos necesarios para manipular las entidades primitivas en programación como paso previo para iniciar el proceso de creación de algoritmos en Python		X	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_2 Identificar los conocimientos necesarios para elaborar algoritmos básicos.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Construir tablas de verdad aplicando los conocimientos adquiridos para entender las respuestas lógicas obtenidas con el computador.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de programación secuenciales para la resolución de problemas		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Emplear estructuras de selección para crear algoritmos que puedan cambiar la secuencia de un proceso.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_3 Cuestionario Lógica de Programación y Algoritmia para validar los conocimientos adquiridos por los aprendices.		x	
GUIA 5 LÓGICA DE PROGRAMACIÓN A	05_4: Aplicar los conceptos de Lógica de Programación y algoritmia en el proyecto seleccionado.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	2.1 El diagrama de clases UML y la Programación Orientada a Objetos: PRESENTACION		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1. ACTIVIDAD DE REFLEXIÓN INICIAL		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.1 Actividad de Aprendizaje 6_1: Interpretar el Diagrama de Clases partiendo del conocimiento previo.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.2 Actividad de Aprendizaje 01_2: Reconocer los componentes del Diagrama de Clases según estándares UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.3 Actividad de Aprendizaje 01_3 Adicionar relaciones al Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.4 Actividad de aprendizaje 01_6 Identificar las clases y restricciones de un Diagrama de Clases.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.5 Actividad de Ejercicios iniciales sobre conceptos de Programación básica en php		X	

GUARDAR

GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.6 Analizar la importancia de la programación orientada a objetos y la importancia de los requisitos para la aplicación.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.7 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.1.8 Actividad de Ejercicios iniciales sobre conceptos de Programación Orientada a Objetos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.1 Actividad de Aprendizaje 06_1 Analizar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.2 Actividad de Aprendizaje 06_2 Interpretar Diagramas de Clases según el modelamiento UML.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.3 Asociar los conceptos relacionados con el patrón de diseño MVC aplicados al desarrollo de aplicaciones web en PYTHON		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.4 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.5 Actividad de Aprendizaje		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.6 Realizar el siguiente ejemplo ubicado en el “taller 06_2_Programacion_Orientada_objetos”		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.7 Realizar el siguiente ejemplo ubicado en el taller 06_03_Programacion_Orientada_objetos2		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.2.8 Realizar el siguiente ejemplo ubicado en el “7_1_Lectura_Entendiendo_MVC_en_PYTHON”,		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.1 Actividad de Aprendizaje 03_1 Determinar los datos e información necesaria para el diseño del Diagrama de Clases		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.2 Actividad de Aprendizaje 06_2 Diseñar Diagramas de Clases para diferentes problemas reales según parámetros establecidos.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.3. Actividades de apropiación del conocimiento (Conceptualización y Teorización). Creando una Aplicación Web Desde Cero Usando Python Flask y MySQL _ Envato Tuts+.		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.4. Actividades de apropiación del conocimiento: Comprender la implementación del patrón Modelo Vista controlador en aplicaciones web Php python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.3.5 Actividades de apropiación del conocimiento: Implementar el patrón Modelo Vista controlador en aplicaciones web Python		X	
GUIA_ 6_DESARR BACKEND EN PYTHO	3.4.1 Actividad de Aprendizaje 04_1 Definir las clases y elementos fundamentales para el diseño del proyecto		X	

[illegible]



SERVICIO NACIONAL DE APRENDIZAJE SENA
CENTRO DE GESTIÓN DE MERCADOS, LOGÍSTICA Y TECNOLOGÍAS DE LA INFORMACIÓN
PLAN DE TRABAJO CONCERTADO
Desarrollo Proceso Formativo

PROGRAMA FORMACIÓN: TEC PROGRAMACIÓN DE SOFTWARE

FICHA: 3432895

INSTRUCTOR: BERNARDO BETANCOURT L. Y LUZ LADY VAZQUEZ

GUÍA DE APRENDIZAJE : Guia 00 Inducción **FECHA:** 13/04/2026

Tabla 1. Descriptores para el Desarrollo de la Ruta de Aprendizaje.						
NOMBRE DEL APRENDIZ	Actividad a desarrollar	Forma de Entrega de Evidencias		Fecha de Entrega	ENTREGO	
		Físico	Digital		SI	NO
ASTRID DAYANA LANCHEROS VIRGUEZ	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
CAROL NICOL GARNICA BETANCURT	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
CRISTHOFR ALCALA BOBADILLA	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
DARA LIZETH TERNERA TOVAR	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
DUBER SANTIAGO CUADROS RINCON	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
EDYSMAR CAROLINA CHIRINOS MATAMOROS	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
IVARELYS PAOLA GONZALEZ ZAMORA	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
JESSICA SOFIA VALENCIA VELASQUEZ	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		

JESSICA JOLIA VALENZUELA VASQUEZ	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
JESUS DANIEL COLINA GONZALEZ	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
JUAN DAVID CALDERON MEDINA	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
JULIAN ANDRES BUITRAGO CULMA	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
JULIAN DAVID TRUJILLO MARIN	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
JULIAN ESTEBAN LOAIZA BARRIOS	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
KAREN DAYANA SUESCA MONTAÑA	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
LEINNY XIOMARA OSPINA DUCUARA	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
LENNYS CAROLINA MOSQUERA MELENDEZ	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
MICHEL YULEY DIAZ ROMERO	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
NIXON DANIEL DURAN MORALES	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
ORBIN STEVEN IZQUIERDO SEGURA	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		
	Actividades_Apropiación_Conocimiento		X	21/04/2026		
	Actividades_Transferencia		X	23/04/2026		
PABLO ANGEL TRUJILLO MARIN	Actividades_Reflexión_Inicial		X	14/04/2026		
	Actividades_Contextualización		X	16/04/2026		

