PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. **IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE:** Validación y entrega del producto al cliente.

- **Denominación del Programa de Formación:** Técnico en Diseño e Integración de Multimedia
- **Código del Programa de Formación:** 524139
- **Nombre del Proyecto: Nombre del Proyecto:** Implementación de solución multimedial que responda a necesidad de comunicación corporativa y académica de la institución educativa.
- **Fase del Proyecto:** Evaluación
- **Actividad de Proyecto:** Realizar la revisión del producto teniendo en cuenta las recomendaciones del cliente.
- **Competencia:** Diseñar la solución multimedial de acuerdo con el informe de análisis de la información recolectada.
- **Resultados de Aprendizaje Alcanzar:** Validar el prototipo con el director del proyecto multimedia de acuerdo con el guion técnico dejando constancia en la lista de chequeo.
- **Duración de la Guía:** 60 horas.

2. PRESENTACIÓN

El diseño de cualquier tipo de productos ya sean mensajes, objetos o espacios, parte siempre de un problema humano, al cual, se le debe dar una solución que satisfaga de manera positiva una necesidad. Durante el desarrollo de la formación, el aprendiz, se ha podido encontrar con una serie de procesos, que le permiten dar solución a una problemática, a partir de las herramientas que ofrece el diseño. Es importante tener siempre en mente un objetivo desde y para el cual, se ofrece una respuesta de diseño adecuada.

En resumen, después de recolectar la información del cliente y organizarla, se idea un producto de diseño que responda a sus necesidades, a continuación, se procede a la bocetación y generación de un prototipo, el cual, se testea y ajusta, cuantas veces sea necesario, para que cumpla de forma positiva con su cometido hasta su producción final, no sin antes, probar que todos los componentes estén presentes y en funcionamiento operativo, para cada una de las tareas que le corresponden, así como también, corregir los errores ortográficos y comprobar que los elementos interactivos estén funcionando correctamente, para lo anterior se hace uso de una lista de chequeo. Por último, se recopilan archivos editables, documentos de los productos realizados y de ser necesario se capacita al cliente en el uso de los elementos para la entrega final.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de reflexión inicial.

El El proceso de producción de diseño e integración de multimedia, por lo general, se compone de una serie de pasos o etapas que van desde la idea o problema de un cliente, hasta la obtención del producto de diseño o multimedia final. La correcta organización del equipo de trabajo en el desarrollo gráfico o audiovisual permite trabajar los múltiples aspectos de nuestra producción en paralelo, ofreciendo alternativas a la línea de producción, lo que es útil para la optimización de tiempos.

De otra parte, cada uno de los procesos que hacen parte de una producción generan archivos y documentos, los cuales, deben conservarse hasta el final de la producción, ya que algunos serán parte integral del producto terminado que se entrega al cliente.

Veamos el siguiente video y preguntémonos ¿Cuál es el pipeline que se aplica al desarrollo de proyectos de diseño o multimedia? https://youtu.be/_zn_1cf87vc.

3.2. Actividades de contextualización.

Veamos el siguiente video “Postproducción" acerca del proceso de finalización de un elemento multimedia, desde la integración de los múltiples elementos que lo conforman hasta su exportación en el formato adecuado: https://youtu.be/kYGmscYBAn4.
Una vez visto el video, responda las siguientes preguntas:

- ¿Cuáles son los pasos que se desarrollan en la fase de postproducción?
- ¿Qué es el montaje?
- ¿Qué programas se pueden utilizar para la postproducción audiovisual?
- ¿Qué tienen en común los programas para postproducción?
- ¿Cuáles son las transiciones principales?

3.3. Actividades de aprendizaje

Actividad: Realizar una reunión con el director del proyecto para la aplicación de pruebas de validación.

I. Actividad de apropiación - interpretación
Consulte en internet acerca del proceso de validación y entrega de un producto terminado.

II. Actividad de argumentación
De manera individual realice al menos 3 tipos de pruebas de usabilidad y navegabilidad para la validación de su producto.

III. Actividad de proposición
En parejas elabore el informe de entrega de su producto multimedia, describiendo detalladamente y justificando el contenido de cada una de las secciones que componen su página web. Para esta actividad cuenta con 8 horas.

3.3 Actividades de transferencia del conocimiento

Según los elementos multimediales realizados para su proyecto productivo, incluidos de cada una de las secciones realizadas para el entregable de la página web elaborada, revise mediante una lista de chequeo que se encuentren activos y en su ubicación adecuada, recuerde aplicar un formato de lista de chequeo para la entrega final de una página web. Teniendo en cuenta los conceptos aprendidos acerca de navegabilidad y usabilidad en la web.

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento: **Evidencias de Desempeño:** **Evidencias de Producto:**	Valida la multimedia de acuerdo con las especificaciones dadas por el director del proyecto. Página Web con elementos multimediales integrados. Informe escrito del correcto funcionamiento del producto multimedial.	VP Valoración de producto LCH Lista de chequeo

5. GLOSARIO DE TÉRMINOS

- **NAVEGABILIDAD**: Facilidad con la que un usuario puede desplazarse por todas las páginas que componen un sitio web. Para lograr este objetivo, un sitio web debe proporcionar un conjunto de recursos y estrategias de navegación diseñados para conseguir un resultado óptimo en la localización de la información y en la orientación para el usuario.
- **USABILIDAD**: El neologismo usabilidad (del inglés usability —facilidad de uso—) se refiere a la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto. La usabilidad también puede referirse al estudio de los principios que hay tras la eficacia percibida de un objeto. La usabilidad es un término que no forma parte del diccionario de la Real Academia Española (RAE), aunque es bastante habitual en el ámbito de la informática y la tecnología.
- **ÚTIL**: Es necesario preguntarnos si nuestros productos y sistemas son útiles, y aplicar nuestro conocimiento para definir soluciones innovadoras que apoyan la utilidad.
- **USABLE**: Corresponde a la facilidad de uso o Usabilidad sigue siendo un aspecto fundamental, necesario, pero no suficiente, por lo que se debe complementar con las demás facetas.
- **DESEABLE**: Si bien los sitios deben ser eficientes, en particular con el uso de medios más complejos (imágenes, sonidos, animaciones), esto se debe equilibrar con los demás valores del diseño emocional.
- **ENCONTRABLE**: Los Sitios Web deben ser navegables y permitir que los usuarios puedan encontrar lo que necesitan.
- **ACCESIBLE**: Los sitios Web deben ser accesibles a las personas con discapacidades (más de 10% de la población). Para los Sitios Web de Gobierno ya es un requisito normativo.
- **CREÍBLE**: La credibilidad es uno de los factores más importantes de tener en cuenta y por ello se deben revisar los elementos de diseño que afectan la confianza que nos tienen los usuarios.
- **VALIOSO**: Las facetas ayudan a determinar los aspectos que llevan a que nuestros sitios ofrezcan valor para nuestros usuarios.

6. REFERENTES BILBIOGRÁFICOS

- IEM Bussiness School Valencia; “La importancia del proceso de investigación y pruebas en el diseño de productos digitales”. http://iembs.com/pruebas-en-productos-digitales/
- Guía Digital; ”¿Qué es la Usabilidad?”. http://www.guiadigital.gob.cl/articulo/que-es-la-usabilidad
- Creativos OnLine, Blog; “Cómo entregar un arte final a un cliente de forma correcta”. https://www.creativosonline.org/blog/como-entregar-un-arte-final-a-un-cliente.html
- Catarsis, Blog; “Malos Entendidos Del Diseño De Experiencia De Usuario”. http://entraencatarsis.com/malos-entendidos-del-diseno-de-experiencia-de-usuario-ux/

7. CONTROL DEL DOCUMENTO

	Nombre	**Cargo**	**Dependencia**	**Fecha**
Autor (es)	Henry Alexander Falla O.	Instructor	Articulación con la media	15 de febrero de 2018
	Juan Carlos Castro G.	Instructor	Articulación con la media	15 de febrero de 2018

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

	Nombre	**Cargo**	**Dependencia**	**Fecha**	**Razón del Cambio**
Autor (es)	Juan Carlos Castro Galvis	Instructor	Articulación con la media	13 abril de 2020	Cambio de Formato SIGA
	Hernan Javier Fierro O.	Instructor	Articulación con la media	13 abril de 2020	Cambio de Formato SIGA
	Susana Páez Urrea	Instructor	Articulación con la media	13 abril de 2020	Cambio de Formato SIGA
	Sergio Alfredo Morales D.	Instructor	Articulación con la media	13 abril de 2020	Cambio de Formato SIGA

	Nelson Fernando Ruiz C.	Instructor	Articulación con la media	10 julio 2023	Cambio de Formato SIGA
	Sergio Alfredo Morales D.	Instructor	Articulación con la media	10 julio 2023	Cambio de Formato SIGA

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Técnico en Programacion de Software
- Código del Programa de Formación: 233104 V.2.0.0
- Nombre del Proyecto: Programación de módulos de sistema de información orientado a servicios
- Fase del Proyecto: Ejecución
- Actividad de Proyecto: Estructurar la base de datos de acuerdo al diseño y el motor de base de datos seleccionado
- Competencia: 220501096 - Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo
- Resultados de Aprendizaje Alcanzar: Codificar el software empleando el lenguaje de programación seleccionado.
- Duración de la Guía: 160 H

2. PRESENTACIÓN



A través de las actividades de esta guía de aprendizaje, **interpretará** algunos ejemplos de código fuente desarrollado empleando las mejores prácticas adoptadas por la industria de Software contrastados con ejemplos de código incorrecto. Esto lo llevará a discutir por qué se necesita estándares, principios y metodologías de codificación. Luego **empleará** las convenciones para nombrar, comentar y dar formato al código fuente, incluidas las clases, los métodos y las variables.

Un programa grande puede ser bastante difícil de entender y mantener. Para los programadores junior, no conocer el código y lo que hace puede ser un aspecto desalentador. Los equipos pueden tener dificultades para trabajar juntos en tales proyectos. Y desde el punto de vista de las pruebas, puede hacer las cosas bastante difíciles. Con el desarrollo de las actividades de la guía de aprendizaje **comprenderá** algunas pautas de diseño de programación, principalmente: KISS, YAGNI, DRY, SOLID y Occam's Razor, entre otras. El proposito primordial de los estándares y principios de codificación es que los programadores mejoren en su oficio programando código que sea más eficiente y fácil de mantener.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión Inicial (antes de Aprender)

3.1.1 Actividad de Reflexión Inicial No.1: Describir los efectos que tienen las malas prácticas de programación en el proceso de desarrollo de software.

A partir de la observación del video "Siete (7) desastres del desarrollo de software | La importancia de la calidad"**,** describa los efectos que tuvieron las malas prácticas de codificación en los proyectos expuestos y los efectos sobre sus destinatarios.

URL: https://youtu.be/6FsmLwUlTV4



- **Estrategia Didáctica**:

  Aprendizaje interactivo.

- **Técnicas Didácticas:**

  Observación, expresión oral, debate.

  Descripción:

  

  El instructor propiciara un debate entre los aprendices a cerca de las consecuencias que tienen las malas prácticas de codificación, la ausencia de control y aseguramiento de la calidad en el proceso de desarrollo de software.

- **Ambiente requerido para la actividad:**

  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.

- **Materiales para la actividad:**

  Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo.

- **Tiempo asignado para el desarrollo de la actividad**

  Tres (3) Horas

3.2 Actividades de Contextualización (antes de Aprender)

3.2.1 Actividad No. 1 de contextualización: - Identificar los errores más comunes que comenten los desarrolladores de software al programar.

A partir de la observación del video "15 Errores que cometes al programar", identificar los errores más comunes que comenten los desarrolladores y el impacto que tienen en los proyectos de software en términos de **tiempo**, **costo** y **alcance.**

URL: https://youtu.be/GoAxsdg0Xbs



- **Estrategia Didáctica**:

  Aprendizaje interactivo.

- **Técnicas Didácticas:**

  Observación, expresión oral, debate.

  Descripción:

  

  Tras la observación del video, el instructor propiciara un debate entre los aprendices presentando proposiciones que ellos deben apoyar o rebatir con la intención de concientizarlos acerca de la importancia de emplear buenas prácticas de programación y las ventajas que ellas ofrecen.

- **Ambiente requerido para la actividad:**

  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.

- **Materiales para la actividad:**

  Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo.

- **Tiempo asignado para el desarrollo de la actividad**

  Tres (3) Horas

- **Entregables y Medio de Entrega.**

  Documento en Microsoft Word con normas APA con por lo menos una página de las conclusiones del debate. LMS Territorium.

3.3 Actividades de Apropiación (Mientras Aprende)

3.3.1 Actividad No. 1 de Apropiación: Conocer los paradigmas de programación sus tipos y enfoques.

Realizar lecturas comprensivas que le permitan conocer los tipos y enfoques de programación disponibles, para lo cual debe remitirse al material de apoyo denominado “MA-PP-Programming Paradigms.md”.

- **Estrategia Didáctica**:

  Auto aprendizaje

- **Técnicas Didácticas:**

  Estudio individual, Elaboración conceptual.

  Descripción:



El Instructor suministrará el material de apoyo MA-PP-Programming Paradigms.md”. Dará las pautas para la elaboración conceptual y la ejemplificación de los conceptos asociados a los enfoques de programación.

- **Material de apoyo:**

  MA-PP-Programming Paradigms.md

- **Ambiente requerido para la actividad:**

  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.

- **Materiales para la actividad:**

  Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo.

- **Tiempo asignado para el desarrollo de la actividad**

  Seis (6) Horas

- **Entregables y Medio de Entrega.**

a) Un (1) mapa mental que represente los siguientes conceptos y sus relaciones a partir de las lecturas realizadas (enfoques de programación). Formato JPG. PNG o PDF. LMS Territorium.

Tipo Imperativo

Enfoques:

- Procedural
- Orientado a objetos.
- Procesamiento en paralelo.

Tipo Declarativo

Enfoques:

- Lógico
- Funcional
- Flujo de Datos (Dataflow)
- Database.

b) Documento en Microsoft Word con normas APA con un ejemplo práctico de cada uno de los enfoques estudiados. Diferente a los contenidos en el material de apoyo. LMS Territorium.

3.3.2 Actividad No. 2 de Apropiación: Interpretar términos base del enfoque de programación Orientada a Objetos (POO)

Empleando herramientas CASE (Visual Parading, Enterprise Architec, Modelio) elaborar diagramas de Clases y Objetos (Notación UML) que faciliten la interpretación de los siguientes conceptos de POO:

a) Clases concretas y sus partes (Nombre, campos, propiedades, métodos).
b) Objetos.
c) Prototipos de objetos en JS.
d) Constructores y Destructores.
e) Encapsulamiento (get y set).
f) Clases estáticas.
g) Visibilidad y Modificacores de acceso.
h) Eventos.

- **Estrategia Didáctica**:

  Exposición.

- **Técnicas Didácticas:**

  Exposiciones del docente.

Descripción:



El Instructor expondrá a los aprendices los conceptos de clases concretas, clases estáticas, visibilidad y modificadores de acceso.
Empleando herramientas CASE, realizara ingeniería inversa para generar el código fuente correspondiente a cada uno de los artefactos UML elaborados.
Recalcara la importancia del modelado para la correcta construcción del código fuente.

- **Material de apoyo:**

  "MA-POO-PROGRAMACIÓN ORIENTADA A OBJETOS"

- **Ambiente requerido para la actividad:**

  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.

- **Materiales para la actividad:**

  Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo.

- **Tiempo asignado para el desarrollo de la actividad**

  Diez (10) Horas

- **Entregables y Medio de Entrega.**

  1. Diagramas de clases en notación UML realizado en herramientas CASE (Visual Parading, Enterprise Architech o equivalente) ejemplificando cada uno de los siguientes conceptos:
     Clase (Constructor, destructor, atributos (Campos y/o Propiedades), Métodos)
     Clase(s) estática(s),
  2. Una tabla comparativa entre los distintos tipos de modificacores de acceso.
  3. Código fuente generado a través del proceso de Ingeniería inversa correspondiente a los diagramas elaborados en el entregable uno(1) de esta Actividad.

Plataforma LMS Territorium. Actividad "Diagramas de clases en notación UML"

3.3.3 Actividad No. 3 de Apropiación: Comprender los tipos de asociación y herencia entre clases en el enfoque de programación Orientada a Objetos (POO)

Empleando herramientas CASE (Visual Parading, Enterprise Architec, Modelio) elaborar diagramas de Clases y Objetos (Notación UML) con distintos tipos de asociación que permitan la comprensión de los siguientes conceptos de POO:

Asosicación Unidireccional.
Asociación Bidireccional.
Herencia.

Clases Abstractas.
Composición.
Agregación.
Interfaces.

- **Estrategia Didáctica**:

  Exposición.

- **Técnicas Didácticas:**

  Exposiciones del docente.

  Descripción:



El Instructor expondrá a los aprendices los conceptos de asociación entre clases, herencia, clases abstractas, composición, agrgación e interfaces.
Empleando herramientas CASE, realizara ingeniería inversa para generar el código fuente correspondiente a cada uno de los artefactos UML elaborados.
Resaltará las ventajas de usar abstracciones y no concreciones para procurar un bajo acoplamiento y una alta cohesión entre clases.

- **Material de apoyo:**

  MA-POO-PROGRAMACIÓN ORIENTADA A OBJETOS"

- **Ambiente requerido para la actividad:**
  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.
- **Materiales para la actividad:**
  Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo.
- **Tiempo asignado para el desarrollo de la actividad**
  20 Horas
- **Entregables y Medio de Entrega.**
  a) Un diagrama de clases que represente cada uno de los siguientes tipos de asociación: Asociación Unidireccional. Asociación Bidireccional. Herencia. Composición. Agregación.
  b) Código fuente generado a través del proceso de Ingeniería inversa correspondiente a los diagramas elaborados en el entregable uno(1) de esta Actividad.

Nota: El Diagrama debe ser construido con herramientas CASE (Visual Parading, Enterprise Architech, etcétera).

Plataforma LMS Territorium. Actividad Diagrama de clases y tipos de asociación.

3.3.4 Actividad No. 4 de Apropiación (Interpretativa): Resolver problemas simples de programación Orientada a Objetos

En el entorno de programación Node JS desarrolle el código fuente que dé solución a los ejercicios propuestos en el material de apoyo denominado: MA_Práctica_ POO1_Guía1_AA3.3.4

- **Estrategia Didáctica**:

  Método basado en problemas.

- **Técnicas Didácticas:**

  Situaciones problema.

  Descripción:



  El Instructor ayudara a los aprendices a reflexionar e identificar necesidades de información y los motivará a resolver cada uno de los problemas planteados.

  Sin embargo, no es un observador pasivo, debe estar orientando el proceso de aprendizaje asegurándose de que el grupo no pierda el objetivo trazado, y de que identifiquen los temas pertinentes de POO para que cumplan con la solución de los problemas.

- **Material de apoyo:**

  MA_Práctica_ POO1_Guía1_AA3.3.4

- **Ambiente requerido para la actividad:**

  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.

- **Materiales para la actividad:**
  Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo. IDE de programación (Node, Express, Mongo DB, React)

- **Tiempo asignado para el desarrollo de la actividad**
  30 Horas

- **Entregables y Medio de Entrega.**

  Código fuente desarrollado en el entorno Node JS para cada ejercicio propuesto en la práctica (Práctica_ POO1_Guía1_AA3.3.4)

Plataforma LMS Territorium. Práctica_ POO1_Guía1_AA3.3.4

3.3.5 Actividad 5 de Apropiación (Interpretativa)

Nombre de la actividad: Construir una API REST empleando Node JS, Mongo DB y Express.

Descripción:

En el entorno de programación Node JS construya una API REST para los recursos ***CURSOS*** y ***USUARIOS*** referidos en el laboratorio (AA-3.3.5-Laboratorio-Construcción de una API REST).

Construya una base de datos Not Only SQL orientada a documentos en Mongo DB.

Para el control de versiones utilice un repositorio de código fuente (Github). Implemente la estrategia GitFlow.

Construya la documentación de la API REST empleando Swagger o Postman.

Despliegue la API en un ambiente Cloud (Heroku, Azure, AWS, IBM Cloud, Oracle Cloud, Huawei Cloud).

- **Estrategia Didáctica**:

  Método basado en situaciones o casos.

- **Técnicas Didácticas:**

  Estudio de Caso. Prácticas Guiadas.

  Descripción:



El Instructor ayudara a los aprendices a reflexionar e identificar necesidades de información y los motivará a resolver cada uno de los requerimientos planteados en el caso de estudio.

Sin embargo, no es un observador pasivo, debe estar orientando el proceso de aprendizaje asegurándose de que el grupo no pierda el objetivo trazado, y de que identifiquen los temas pertinentes de POO para que cumplan con la solución de los problemas.

- **Material de apoyo:**

MA_Instructivo1_Guía1_AC.3.3.5. "*Construcción de una API REST con NodeJS, Express y MongoDB" (Cortes Arévalo, 2022, p. 1 - 59).*

- **Ambiente requerido para la actividad:**

  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.

- **Materiales para la actividad:**

Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo. IDE de programación (Node, Express, Mongo DB, React)

- **Tiempo asignado para el desarrollo de la actividad**

  60 Horas

- **Entregables y Medio de Entrega.**

  1) URL del Repositorio de código fuente de la API REST que da solución al caso de estudio Alquiler de Motocicletas..
  2) Un video a través del cual se evidencie el funcionamiento de los EndPoint de cada uno de los recursos expuesto en la API. Debe usar POSTMAN para demostrar el funcionamiento de los End Point en el ámbito local y remoto (Producción).
  3) Documentación generada por Postman de los End Point probados.

  Territorium. Actividad “Link de acceso a repositorio público API REST en Node JS Motocicletas”.

3.4 Actividades de Transferencia del conocimiento (Después de aprender):

Nombre de la actividad: Desarrollar el Backend que dé solución al caso de estudio Motocicletas. Véase material de apoyo “MA-CE-CASO DE ESTUDIO MOTORCYCLE COMPANY”.

Descripción:

Desarrollar el Backend que de solución al caso de estudio empleando el Stack de Programación MEAN, el paradigma de APIS “Request Response” y el tipo de API “REST” para cada uno de los recursos del problema abordado.

- **Estrategia Didáctica**:

  Método Masado en Proyectos.

- **Técnicas Didácticas:**

  Estudio de Caso. Prácticas Guiadas.

  Descripción:



El Instructor ayudara a los aprendices a reflexionar e identificar necesidades de información y los motivará a resolver cada uno de los requerimientos del proyecto formativo..

Sin embargo, no es un observador pasivo, debe estar orientando el proceso de aprendizaje asegurándose de que el grupo no pierda el objetivo trazado, y de que identifiquen los temas pertinentes de POO para que cumplan con la solución de los problemas.

- **Material de apoyo:**

MA_Instructivo1_Guía1_AC.3.3.5. "*Construcción de una API REST con NodeJS, Express y MongoDB" (Cortes Arévalo, 2022, p. 1 - 59).*

- **Ambiente requerido para la actividad:**
  Ambiente de formación, Ayudas audiovisuales (Televisor o Video Beam, conectividad a Internet alámbrica o inalámbrica), mesas, sillas.
- **Materiales para la actividad:**
  Computador de escritorio o portátil, sistema operativo Windows (recomendable), software ofimático y material de apoyo. IDE de programación (Node, Express, Mongo DB, React)
- **Tiempo asignado para el desarrollo de la actividad**
  28 Horas
- **Entregables y Medio de Entrega.**
  1) Código fuente de la API REST asociada al proyecto formativo

     El Backend debe estar codificado en NODE JS, debe compilar 100% y se deben evidenciar buenas prácticas de programación.

     El código Fuente debe estar en un Repositorio Público tipo Github. Se debe evidenciar buenas practicas de integración de código, a saber: Git Semántico. GitFlow.

  2) Debe realizar un video a través del cual se evidencie el funcionamiento de los EndPoint de cada uno de los recursos expuesto en la API. Debe usar POSTMAN para demostrar el funcionamiento de los End Point en el ámbito local y remoto (Producción).
  3) Documentación generada por Postman de los End Point probados.
  4) URL de los End Point Bublicados (Producción)

  Territorium. Actividad "Solución al Proyecto Formativo".

4. ACTIVIDADES DE EVALUACIÓN

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

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : Codificar la API REST de acuerdo a los requerimientos de software del caso de estudio “Alquiler de motocicletas” **Evidencias de Desempeño** Despliegue de la API REST en ambiente Cloud (Heroku) **Evidencias de Producto:** Construcción de la documentación de la API REST	Utiliza herramientas y entornos de desarrollo, para la codificación de los módulos del sistema, aplicando la sintaxis propia del lenguaje de programación seleccionado, de acuerdo con las necesidades del sistema de información. Construye un repositorio de código fuente para el versionamiento del proyecto de software. Crea la base de datos en el motor de base de datos seleccionado, siguiendo especificaciones técnicas de del informe, según normas y protocolos de la empresa.	**Lista de chequeo IE-** AA.3.3.5.a. Subactividad 1: Codificar la API REST de acuerdo a los requerimientos de software. **Lista de chequeo IE-** AA.3.3.5.b. Subactividad 2: Desplegar la API REST en ambiente Cloud (Heroku, AWS, AZURE). **Lista de chequeo IE-** AA.3.3.5.c Subactividad 3: Construir la documentación de la API REST.

5. GLOSARIO DE TÉRMINOS

API: traduce interfaz de programación de aplicaciones. Este recurso permite que productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementado.
Base de Datos: Es una colección de datos organizados de forma ordenada y almacenada sistemáticamente para operaciones posteriores.

JSON: JavaScript Object Notation (JSON) Es un formato basado en texto estándar que se utiliza para representar datos estructurados en la sintaxis de objetos de JavaScript. Se utiliza principalmente para transmitir datos en aplicaciones web (como es el caso de enviar algunos datos desde el servidor al cliente, por lo tanto, estos datos pueden ser mostrados en páginas web, o viceversa).

Localhost: en el contexto de las redes de computadoras localhost es una computadora que ejecuta un programa. La computadora que se tiene al frente funciona también como un servidor virtual; este modo es comúnmente utilizado para realizar pruebas en el desarrollo web.

Manejadores o handlers: de eventos: en la programación con JavaScript un evento por sí mismo carece de utilidad; para que los eventos resulten útiles o aplicables se debe asociar funciones o código JavaScript a cada evento. De esta forma, cuando se produce un evento de manera inmediata se ejecuta el código indicado,

por lo que la aplicación responde ante cualquier evento que se produzca durante su ejecución. Las funciones o código JavaScript que se realizan para cada evento se denominan "manejador de eventos".

Middleware: es un bloque de código que se ejecuta entre la petición que hace el usuario (request) hasta que esa petición llega al servidor. De esta manera son funciones que tienen acceso al objeto de solicitud (req), el objeto de respuesta (res) y la siguiente función de middleware en el ciclo de solicitud-respuesta de la aplicación.

Servidor Web: Una computadora de grandes especificaciones hardware que aloja un sitio web o plataformas con todos sus recursos en la Internet.

Servlet: es una clase hecha en Java utilizada para extender las capacidades de los servidores que alojan aplicaciones que siguen el modelo petición-respuesta

6. REFERENTES BILBIOGRÁFICOS

BIBLIOGRAFÍA

¿Qué es una API? (31 de Octobre de 2017). Obtenido de RedHat: https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces

and, J. A. (2021). *C# 9.0 Pocket Reference.* O'Reilly Media, Inc.

Antani, V. (2017). *Object-Oriented JavaScript - Third Edition.* Packt Publishing.

Bouna, P. (2022). *C# Object-Oriented Programming for Beginners in C# and .NET.* Packt Publishing.

Gabriel Baptista, F. A. (2022). *Software Architecture with C# 10 and .NET 6 - Third Edition.* Packt Publishing.

Herron, D. (2020). *Node.js Web Development.* Packt Publisind.

Kereki, F. (2018). *Modern JavaScript Web Development Cookbook.* Packt Publising.

Laurence Svekis, M. v. (2021). *JavaScript from Beginner to Professional.* Packt Publising.

Padolsey, J. (2020). *Clean Code in JavaScript.* Packt Publising.

Wexler, J. (2019). *Get Programming with Node.js.* Manning Publications.

CIBERGRAFÍA

¿Qué es una API? (2017, Octobre 31). Retrieved from RedHat: https://www.redhat.com/es/topics/api/what-are-application-programmin (Herron, 2020)g-interfaces

Choque, J. C. (2016). Adición de estándares de codificación node.js. GitLab https://gitlab.agetic.gob.bo/dbarra/bolivia-libre/-/commit/1ff84b6ae70f0e9436f6e6ddc7f3e060bc9556ad

Capacho Portilla, J. R., Nieto Bernal, W.(2017). Diseño de base de datos. Editorial Universidad del Norte. https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=9740

Cortes Arévalo, V. C. V. (2022, abril). Construcción de una API REST con NodeJS, Express y MongoDB.

Coronel, C., Rob, P., Morris, S.(2011). Bases de datos: diseño, implementación y Administración. Cengage Learning. https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=1280

Express. (2022). Cómo empezar. https://expressjs.com/es/starter/installing.html

López, A. (2021). Qué es Postman y para qué sirve. OpenWebinars https://openwebinars.net/blog/que-es-postman/

Mozilla. (2022). Await. MDN Web Docs. https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/await

Mozilla. (2022). Introducción a Express/Node. MDN Web Docs. https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

Nodejs.org. (2022). Acerca de Node.js. Node.js https://nodejs.org/es/about

npm. (2022). About npm. https://www.npmjs.com/about

7. CONTROL DEL DOCUMENTO

	Nombre	**Cargo**	**Dependencia**	**Fecha**
Autor (es)	VÍCTOR CLAUDIO VLADIMIR CORTÉS ARÉVALO	INSTRUCTOR GRADO 20	Centro de Electricidad, Electrónica y Telecomunicaciones.	15-noviembre-2021

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

	Nombre	**Cargo**	**Dependencia**	**Fecha**	**Razón del Cambio**
Autor (es)					