

## UNIVERSIDAD TECNOLÓGICA DE HERMOSILLO

### PLANEACIÓN DIDÁCTICA DEL PROGRAMA DE ASIGNATURA

(1) Programa educativo:	LICENCIATURA EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN E INNOVACIÓN DIGITAL	(2) Docente (s):	BERNARDO PRADO DIAZ
(3) Cuatrimestre:	5°	(4) Periodo escolar:	enero – abril 2026
(5) Nombre de la asignatura:	APLICACIONES WEB ORIENTADAS A SERVICIOS	(6) Grupo (s):	5-1 5-2

(7) Propósito de la asignatura:	El estudiante desarrollará aplicaciones web con acceso a base de datos implementando programación orientada a objetos para su publicación y distribución en internet.
---------------------------------	---

(8) Competencia a la que contribuye la asignatura:	Desarrollar soluciones tecnológicas multiplataforma de software web y móvil utilizando programación orientada a objetos, frameworks, bases de datos, estándares de calidad y diseño para resolver problemas del sector productivo, con un enfoque de inclusión, compromiso con la responsabilidad social, equidad social y de género, excelencia, vanguardia, innovación social e interculturalidad.
--	--

(9) Tipo de competencia	Específica	(10) Créditos	5.63	(11) Modalidad	Escolarizada
-------------------------	------------	---------------	------	----------------	--------------

(12) Horas del saber	36	(13) Horas del saber hacer	54	(14) Horas Totales	90	(15) Horas por semana	6
----------------------	----	----------------------------	----	--------------------	----	-----------------------	---

**INFORMACIÓN DE LA UNIDAD DE APRENDIZAJE** (Agregar esta sección por cada una de las unidades de aprendizaje)

<b>(16) Nombre de la unidad de Aprendizaje:</b>	I. Introducción al desarrollo web orientado a servicios		
<b>(17) Propósito esperado:</b>	El estudiante identificará la arquitectura para el desarrollo de aplicaciones orientadas a servicios.		
<b>(18) Horas del saber</b>	<b>(19) Horas del Saber Hacer</b>	<b>(20) Horas Totales</b>	<b>(21) Porcentaje de la Unidad (Para evaluación sumativa)</b>
<b>6</b>	<b>6</b>	<b>12</b>	<b>25 %</b>

<b>(22) Temas</b>	<b>(23) Saber Dimensión conceptual</b>	<b>(24) Saber Hacer Dimensión actuacional</b>	<b>(25) Saber Ser-convivir Dimensión socioafectiva</b>
Paradigma del desarrollo de aplicaciones orientadas a servicios	Distinguir los servicios que se ofrecen en la nube. Identificar las características de las aplicaciones orientadas a servicios. Identificar el concepto y las características de las aplicaciones Web híbridas (Mashup).	Documentar las ventajas de la arquitectura orientada a servicios.	Fomentar la habilidad de investigación de manera individual y en equipo en la selección de los paradigmas del desarrollo de aplicaciones orientadas a servicios para distinguir cada uno de estos. Promover la gestión de información de la arquitectura orientada a servicios para identificar los procesos de buenas prácticas de diseño para mejorar la calidad del desarrollo de los diagramas de aplicaciones en la arquitectura SOA.
Arquitectura orientada a servicios (SOA)	Definir la arquitectura orientada a servicios. Identificar los principios de diseño que se aplican a sobre cada servicio modelado. Identificar los estándares relacionados a	Desarrollar diagramas de aplicaciones orientadas a servicios que integren los elementos de la arquitectura SOA.	

	los servicios: XML, SOAP, WSDL, UDDI, REST.	
--	---	--

\*Se agrega una fila a esta tabla por cada tema de la unidad.

PROCESO DE EVALUACIÓN POR UNIDAD TEMÁTICA					
(26) Periodo en semanas	1	(28) Evidencia de aprendizaje	(29) Tipo de evaluación	(30) Ponderación %	(31) Instrumento de evaluación
Los estudiantes desarrollan diagramas de aplicaciones orientadas a servicios bajo la arquitectura SOA que incluya las cuatro capas: <ul style="list-style-type: none"> <li>• La capa de acceso.</li> <li>• La capa de procesos.</li> <li>• La capa de servicio.</li> <li>• La capa de recursos.</li> </ul>	A partir de una investigación con tutoría analizar y construir un organizador gráfico de los diferentes diagramas de aplicaciones orientada a servicios.  A partir de una investigación con tutoría analizar y desarrollar un reporte que incluya las características de las aplicaciones Web orientadas a servicios.  A partir de un aprendizaje basado en problemas/soluciones desarrollar un reporte que incluya el procedimiento de la arquitectura orientada a servicios (SOA).	Heteroevaluación (del docente al estudiante). Formativa y sumativa.	100%. De La unidad.	Ejercicios prácticos Lista de verificación	

\*Se deben agregar tantas filas como evidencias haya.

#### SECUENCIA DIDÁCTICA (Agregar esta sección por cada una de las unidades de aprendizaje)

(32) APERTURA			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Lluvia de ideas, Preguntas detonadoras, Exposición guiada	Introduce los conceptos clave de SOA, servicios web y Mashup. Pregunta sobre el uso de APIs en la vida cotidiana.	Participan en la lluvia de ideas. Toman notas de los conceptos clave. Responden al diagnóstico inicial de conocimientos.	Diagnóstico de conocimientos previos (conceptos de servicios y Web).	Pizarra, computadora, cañón proyector, presentación sobre SOA/Mashups, videos sobre arquitectura de servicios.

\*Se agregan tantas filas como se requiera.

(33) DESARROLLO			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		

Estudio de casos, Trabajo colaborativo, Demostración práctica	Explica los componentes y capas de la arquitectura SOA (Acceso, Procesos, Servicio, Recursos). Guía en el análisis de un caso práctico para diagramar el flujo de servicios.	Analizan el caso de estudio. En equipos, desarrollan diagramas UML que representen las cuatro capas de la arquitectura SOA.	Elaboración de diagramas UML de una aplicación SOA con sus 4 capas.	Computadora, Software de modelado (Draw.io, PlantUML), Guías impresas de ejercicios de modelado, Acceso a Internet.
---	--	---	---	---

\*Se agregan tantas filas como se requiera.

(34) CIERRE Desarrollo de la estrategia enseñanza aprendizaje			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Reflexión individual, Discusión en plenaria	Solicita a los equipos presentar sus diagramas de SOA y los reportes de investigación. Proporciona retroalimentación formativa.	Exponen su trabajo y argumentan las decisiones de diseño. Entregan el portafolio de evidencias con el reporte, organizador gráfico y diagramas.	Portafolio de evidencias: Organizador gráfico, Reporte de características de Web Apps orientadas a servicios, Reporte del procedimiento SOA.	Proyector, Rúbrica de evaluación, Lista de verificación.

\*Se agregan tantas filas como se requiera.

### (35) Referencias bibliográficas y digitales:

Referencias Bibliográficas y Digitales (Unidad I)

- Erl, T. (2004). Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). El lenguaje unificado de modelado UML. Pearson Educación.
- Sommerville, I. (2011). Ingeniería de software (9.<sup>a</sup> ed.). Pearson.
- <https://www.uml.org/>
- <https://restfulapi.net/>

**INFORMACIÓN DE LA UNIDAD DE APRENDIZAJE** (Aregar esta sección por cada una de las unidades de aprendizaje)

<b>(16) Nombre de la unidad de Aprendizaje:</b>	II. Implementación de interfaces de programación de aplicaciones de terceros (APIS)		
<b>(17) Propósito esperado:</b>	El estudiante utilizará las API's para programar aplicaciones orientadas a servicios.		
<b>(18) Horas del saber</b>	<b>(19) Horas del Saber Hacer</b>	<b>(20) Horas Totales</b>	<b>(21) Porcentaje de la Unidad (Para evaluación sumativa)</b>
10	20	30	25 %

<b>(22) Temas</b>	<b>(23) Saber Dimensión conceptual</b>	<b>(24) Saber Hacer Dimensión actuacional</b>	<b>(25) Saber Ser-convivir Dimensión socioafectiva</b>
Geolocalización Identificar las API's en el intercambio de	Geolocalización Identificar las API's en el intercambio de	Programar aplicaciones con API's de geolocalización.	Cultivar la capacidad de razonamiento crítico para identificar cada una de las apis utilizadas en los servicios web.
Redes sociales Identificar las API's en el intercambio de	Identificar las API's en el intercambio de información entre aplicaciones de redes sociales. Identificar los elementos del lenguaje de programación necesarios para el uso de API's de redes sociales	Programar aplicaciones con API's de redes sociales.	Incentivar la creatividad al explorar diferentes enfoques en el intercambio de información con las diferentes API's que utilizan los servicios web
Bases de datos Identificar las API's en el intercambio de	Identificar las API's en el intercambio de información entre aplicaciones de bases de datos. Identificar los elementos del lenguaje de programación necesarios para el uso de API's de bases de datos.	Programar aplicaciones con API's de bases de datos.	

Plataformas online	Identificar las API's en el intercambio de información entre aplicaciones de plataformas on line. Identificar los elementos del lenguaje de programación necesarios para el uso de	Programar aplicaciones con API's de plataformas online.
Plataformas streaming	Identificar las API's en el intercambio de información entre aplicaciones de plataformas streaming. Identificar los elementos del lenguaje de programación necesarios para el uso de plataformas streaming.	Programar aplicaciones con API's de plataformas streaming

\*Se agrega una fila a esta tabla por cada tema de la unidad.

PROCESO DE EVALUACIÓN POR UNIDAD TEMÁTICA				
(26) Periodo en semanas	(28) Evidencia de aprendizaje	(29) Tipo de evaluación	(30) Ponderación %	(31) Instrumento de evaluación
<b>(26) Periodo en semanas</b> 2 - 4	<b>(28) Evidencia de aprendizaje</b> Los estudiantes identifican la aplicación y uso de de API's de: <ul style="list-style-type: none"> <li>• Geolocalización.</li> <li>• Redes sociales.</li> <li>• Bases de datos.</li> <li>• Plataformas on line.</li> <li>• Plataformas streaming en aplicaciones web orientadas a servicios</li> </ul>	<b>(29) Tipo de evaluación</b> A partir de aprendizaje basado en problemas/soluciones conformará un portafolio de evidencias que contenga los entregables de las fases de análisis, diseño, desarrollo y pruebas de las siguientes API's: <ul style="list-style-type: none"> <li>• Geolocalización.</li> <li>• Redes sociales.</li> <li>• Bases de datos.</li> <li>• Plataformas on line.</li> <li>• Plataformas streaming.</li> <li>• Funcionamiento de las API's.</li> </ul>	<b>(30) Ponderación %</b> Heteroevaluación (del docente al estudiante). Formativa y sumativa. 100% de la Unidad.	<b>(31) Instrumento de evaluación</b> Ejercicios prácticos

\*Se deben agregar tantas filas como evidencias haya.

#### SECUENCIA DIDÁCTICA (Aregar esta sección por cada una de las unidades de aprendizaje)

(32) APERTURA			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Desarrollo de la estrategia enseñanza-aprendizaje				
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Lluvia de ideas, Análisis de ejemplos, Discusión dirigida	Introduce el concepto de API de terceros (Third-party API) y su uso. Presenta ejemplos de APIs de Geolocalización, Redes Sociales, y Bases de Datos.	Participan en lluvia de ideas. Identifican las APIs de servicios web más comunes. Responden a un cuestionario diagnóstico sobre el manejo de peticiones HTTP.	Cuestionario diagnóstico sobre el manejo de peticiones y respuestas HTTP.	Pizarra, cañón, Presentaciones sobre tipos de APIs, Ejemplos de APIs populares (Google Maps, Twitter).

\*Se agregan tantas filas como se requiera.

..

(33) DESARROLLO			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Desarrollo de la estrategia enseñanza aprendizaje				
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		

Aprendizaje basado en problemas	Muestra cómo consumir una API externa utilizando Python/Django. Guía la instalación de librerías para peticiones HTTP (ej. requests). Propone desafíos progresivos para cada tipo de API (Geo, Redes, BD, Streaming)	Codifican y prueban peticiones a diferentes APIs (simuladas o reales). Desarrollan funciones para procesar e interpretar los datos recibidos (JSON/XML).	Desarrollo de funciones en Python para consumir y procesar datos de APIs de Geolocalización y Redes Sociales.	Computadora con IDE (Visual Studio Code), Lenguaje de programación Python, librerías de red, Acceso a Internet, Documentación de APIs (ej. Google Maps API Key).
---------------------------------	--	--	---	--

\*Se agregan tantas filas como se requiera.

(34) CIERRE			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Desarrollo de la estrategia enseñanza aprendizaje				
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Coevaluación, Discusión en plenaria, Retroalimentación formativa	Evalúa el progreso de los ejercicios prácticos. Fomenta la reflexión sobre los desafíos de seguridad y autenticación al usar APIs.	Presentan sus soluciones y explican cómo implementaron las conexiones a las diferentes APIs. Participan en la coevaluación del código de sus compañeros.	Portafolio de evidencias con los entregables de análisis, diseño, desarrollo y pruebas de las APIs.	Proyector, Rúbrica de evaluación, Checklists de funcionalidad de las APIs.

\*Se agregan tantas filas como se requiera.

### (35) Referencias bibliográficas y digitales:

Referencias Bibliográficas y Digitales (Unidad II) :

Documentación oficial de Python: <https://docs.python.org>

Documentación de la librería Requests (Python): <https://docs.python-requests.org/en/master/>

O'Reilly Media. (2018). APIs For Dummies. Wiley.

Documentación de APIs de terceros (ej. Google Maps API, Twitter Developer Documentation).

**INFORMACIÓN DE LA UNIDAD DE APRENDIZAJE** (Aregar esta sección por cada una de las unidades de aprendizaje)

<b>(16) Nombre de la unidad de Aprendizaje:</b>	III. Desarrollo de una interfaz de programación de aplicaciones (API).		
<b>(17) Propósito esperado:</b>	El estudiante utilizará frameworks para desarrollar aplicaciones web que permitan consumir servicios para satisfacer las necesidades del cliente.		
<b>(18) Horas del saber</b>	<b>(19) Horas del Saber Hacer</b>	<b>(20) Horas Totales</b>	<b>(21) Porcentaje de la Unidad (Para evaluación sumativa)</b>
18	18	36	25 %

<b>(22) Temas</b>	<b>(23) Saber Dimensión conceptual</b>	<b>(24) Saber Hacer Dimensión actuacional</b>	<b>(25) Saber Ser-convivir Dimensión socioafectiva</b>
Frameworks para el desarrollo de interfaces de programación de aplicaciones	Identificar frameworks en el desarrollo de aplicaciones orientadas a servicios.	Programar aplicaciones que utilicen API's para conexión a servicios utilizando un framework de desarrollo.	Fomentar la habilidad de investigación de manera individual y en equipo en la selección de los framework de desarrollo de interfaces de programación de aplicaciones.
Desarrollo de una interfaz de programación de aplicaciones (API)	Identificar el proceso de desarrollo de programación de aplicaciones (API)	Desarrollar interfaces de programación de aplicaciones (API)	Promover la gestión de información de en equipos de desarrollo identificando los procesos de buenas prácticas de programación de aplicaciones.

\*Se agrega una fila a esta tabla por cada tema de la unidad.

PROCESO DE EVALUACIÓN POR UNIDAD TEMÁTICA				
(26) Periodo en semanas	(27) Resultado de aprendizaje de la unidad			
(28) Evidencia de aprendizaje	(29) Tipo de evaluación	(30) Ponderación %	(31) Instrumento de evaluación	
Los estudiantes comprenden la aplicación y uso de los frameworks para el desarrollo de interfaces de programación de aplicaciones (API's)	<p>A partir de aprendizaje basado en problemas/soluciones conformará un portafolio de evidencias que contenga:</p> <ul style="list-style-type: none"> <li>• Lenguaje de programación y el Framework utilizado para el desarrollo.</li> <li>• Elementos que componen una aplicación Web orientada a servicios.</li> <li>• API's desarrolladas en el framework seleccionado</li> </ul>	Heteroevaluación (del docente al estudiante). Formativa y sumativa.	100%.de la Unidad.	De Ejercicios prácticos

\*Se deben agregar tantas filas como evidencias haya.

#### SECUENCIA DIDÁCTICA (Aregar esta sección por cada una de las unidades de aprendizaje)

(32) APERTURA			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Desarrollo de la estrategia enseñanza-aprendizaje		Actividades de estudiantes		
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Tormenta de ideas, Preguntas guiadas	Aplica preguntas diagnósticas sobre Frameworks Web (ej. ¿Qué es Django?). Conecta el contenido con	Participan en la reflexión inicial. Comparten conocimientos previos sobre Django y MySQL y su arquitectura.	Diagnóstico sobre conocimientos en Frameworks Web (Django) y bases de datos relacionales (MySQL).	Pizarra, Presentaciones sobre arquitectura MVC/MVT (Django), Encuestas previas.

	<b>la necesidad de crear APIs propias para la aplicación.</b>			
--	---	--	--	--

\*Se agregan tantas filas como se requiera.

<b>(33) DESARROLLO</b>			<b>Evidencia de aprendizaje</b>	<b>Medios y Materiales didácticos (Recursos)</b>
<b>Desarrollo de la estrategia enseñanza aprendizaje</b>		<b>Actividades de estudiantes</b>		
<b>Métodos y técnicas de enseñanza-aprendizaje</b>	<b>Actividades docentes</b>			
Aprendizaje basado en problemas, Desarrollo colaborativo	Muestra la configuración inicial de un proyecto Django. Guía la implementación del modelo, vistas y serializers para crear endpoints RESTful (API). Demuestra la conexión a MySQL y la realización de consultas.	Crean la estructura del proyecto Django. Programan los endpoints de su API. Realizan pruebas unitarias para garantizar su funcionamiento y conexión a MySQL.	Desarrollo de una API RESTful en Django/Django REST Framework conectada a la base de datos MySQL.	Computadora con IDE (VS Code), Django/Django REST Framework, MySQL Server, Conector de MySQL para Python, Manuales de Django.

\*Se agregan tantas filas como se requiera.

<b>(34) CIERRE</b>		<b>Evidencia de aprendizaje</b>	<b>Medios y Materiales didácticos</b>
<b>Desarrollo de la estrategia enseñanza aprendizaje</b>			

Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		(Recursos)
Revisión entre pares, Retroalimentación, Reflexión final	Retroalimenta el código de las APIs desarrolladas. Muestra cómo documentar los endpoints de manera clara (ej. Swagger/OpenAPI).	Mejoran el código de su API. Crean la documentación técnica de los endpoints y el manual de instalación del proyecto.	Portafolio de evidencias que contenga: Lenguaje y Framework utilizado (Django), Elementos de la Web App, y las APIs desarrolladas.	Rúbricas de revisión de código, Herramientas de documentación de APIs (Postman/Swagger).

\*Se agregan tantas filas como se requiera.

#### (35) Referencias bibliográficas y digitales:

Referencias Bibliográficas y Digitales (Unidad III) :

Documentación oficial de Django: <https://docs.djangoproject.com/>

Documentación de Django REST Framework: <https://www.django-rest-framework.org/>

Documentación de MySQL: <https://dev.mysql.com/doc/>

Martin, R. C. (2009). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.

**INFORMACIÓN DE LA UNIDAD DE APRENDIZAJE** (Aregar esta sección por cada una de las unidades de aprendizaje)

<b>(16) Nombre de la unidad de Aprendizaje:</b>	IV. Implementación		
<b>(17) Propósito esperado:</b>	El estudiante integrará una aplicación orientada a servicios para satisfacer las necesidades del cliente.		
<b>(18) Horas del saber</b>	<b>(19) Horas del Saber Hacer</b>	<b>(20) Horas Totales</b>	<b>(21) Porcentaje de la Unidad (Para evaluación sumativa)</b>
<b>2</b>	<b>10</b>	<b>12</b>	<b>25 %</b>

<b>(22) Temas</b>	<b>(23) Saber Dimensión conceptual</b>	<b>(24) Saber Hacer Dimensión actuacional</b>	<b>(25) Saber Ser-convivir Dimensión socioafectiva</b>
Diseño de una aplicación orientada a servicios	Identificar el proceso de diseño de una aplicación orientada a servicios.	Identificar el proceso de diseño de una aplicación orientada a servicios.	Cultivar la capacidad de integración en la implementación de una aplicación orientada a servicios.
Integración de una aplicación orientada a servicios	Identificar el proceso de integración de una aplicación orientada a servicios.	Integrar aplicaciones Web orientadas a servicios.	Incentivar la creatividad al programar, e implementar una aplicación orientada a servicios en la nube.

\*Se agrega una fila a esta tabla por cada tema de la unidad.

<b>PROCESO DE EVALUACIÓN POR UNIDAD TEMÁTICA</b>				
<b>(26) Periodo en semanas</b>	<b>9 - 10</b>			
<b>(27) Resultado de aprendizaje de la unidad</b>	<b>(28) Evidencia de aprendizaje</b>	<b>(29) Tipo de evaluación</b>	<b>(30) Ponderación %</b>	<b>(31) Instrumento de evaluación</b>

Los estudiantes desarrollan soluciones de software a través de ambientes automatizados de pruebas para garantizar que los resultados obtenidos sean los definidos en los requerimientos.	A partir de aprendizaje basado en programación orientada a servicios conformar un portafolios de evidencias que contenga: Plan de pruebas Criterios de aceptación Resultados obtenidos de las pruebas Aprobación de la solución	Heteroevaluación (del docente al estudiante). Formativa y sumativa.	100%. De la Unidad.	Ejercicios prácticos
--	--	---	---------------------	----------------------

\*Se deben agregar tantas filas como evidencias haya.

#### SECUENCIA DIDÁCTICA (Aregar esta sección por cada una de las unidades de aprendizaje)

(32) APERTURA			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Lluvia de ideas, Preguntas generadoras, Exposición guiada	Presenta la metodología de pruebas (unitarias, integración). Pregunta sobre la experiencia previa en el despliegue de aplicaciones (Hosting).	Participan en la lluvia de ideas. Responden al cuestionario diagnóstico sobre conceptos de pruebas y despliegue (DevOps).	Cuestionario diagnóstico sobre conocimientos previos de pruebas automatizadas y ambientes de despliegue.	Proyector, Pizarra, Presentación sobre Pruebas de Software y Despliegue.

\*Se agregan tantas filas como se requiera.

(33) DESARROLLO			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Desarrollo de la estrategia enseñanza aprendizaje				

Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Aprendizaje basado en proyectos, Práctica guiada, Estudio de casos	Guía el proceso de Integración de la aplicación (Unidades I, II, III). Demuestra la implementación de pruebas unitarias en Django. Explica el procedimiento de despliegue en PythonAnywhere.	Implementan pruebas unitarias y de integración para garantizar que los resultados cumplan con los requerimientos. Despliegan la aplicación Django/MySQL en PythonAnywhere.	Desarrollo de la aplicación final (Integración de API propia y consumo de APIs de terceros). Implementación de pruebas unitarias/de integración	Computadoras, Software de desarrollo (Django), PythonAnywhere platform, Guías de programación y Despliegue.

\*Se agregan tantas filas como se requiera.

(34) CIERRE Desarrollo de la estrategia enseñanza aprendizaje			Evidencia de aprendizaje	Medios y Materiales didácticos (Recursos)
Métodos y técnicas de enseñanza-aprendizaje	Actividades docentes	Actividades de estudiantes		
Exposición oral, Evaluación formativa, Coevaluación	Coordina las presentaciones de los proyectos finales. Evalúa el proyecto utilizando la rúbrica y los criterios de aceptación.	Exponen la aplicación final funcional (incluyendo el despliegue y las pruebas). Argumentan las decisiones técnicas y los resultados obtenidos.	Presentación final del proyecto funcional con el Portafolio de evidencias: Plan de pruebas, Criterios de aceptación, Resultados de pruebas, y Aprobación de la solución	Computadora, Proyector, Rúbricas de evaluación.

\*Se agregan tantas filas como se requiera.

### (35) Referencias bibliográficas y digitales:

Referencias Bibliográficas y Digitales (Unidad IV) :

Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

Robert C. Martin. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall.

Documentación oficial de PythonAnywhere: <https://www.pythonanywhere.com/>

Guías de Pruebas de Django: <https://docs.djangoproject.com/en/stable/topics/testing/>