



1. Datos de identificación:	
Nombre la institución:	Universidad Autónoma de Nuevo León
Nombre de la dependencia:	Facultad de Contaduría Pública y Administración
Nombre del programa educativo:	Licenciado en Tecnologías de Información
Nombre de la unidad de aprendizaje:	Estructuras de datos y su procesamiento
Horas aula-teoría y/o práctica, totales:	100
Frecuencias aula por semana:	5
Horas extra aula, totales:	50
Tipo de modalidad:	Escolarizada
Tipo de periodo académico:	Tercer semestre
Tipo de unidad de aprendizaje:	Obligatoria
Área curricular:	Formación básica (ACFB)
Créditos UANL:	5
Fecha de elaboración:	07/06/19
Fecha de última actualización:	Noviembre 2019
Responsable (s) del diseño y rediseño:	Dr. Francisco Salazar

2. Presentación:

*La Unidad de aprendizaje “Estructuras de datos y su procesamiento” aportará al estudiante la competencia de analizar, diseñar e implementar estructuras de datos en memoria que tengan como finalidad el más eficiente almacenamiento del estado de una solución de negocios; así como su segmentación, manipulación, procesamiento y obtención de información. Lo anterior se logrará en la **primera fase** mediante el aplicar las estructuras nativas del lenguaje Python tales como Listas, Tuplas, Conjuntos y Diccionarios para almacenar en memoria principal el estado de una aplicación a desarrollar y emplearlas para codificar pilas y colas que permitan el proceso de conjuntos de datos, los cuales sentarán las bases para aprovechar estructuras de datos no nativas y provistas mediante librerías. En la **segunda fase** se adquirirá la competencia para diseñar e implementar soluciones de software que aprovechen las estructuras de datos multidimensionales bajo las librerías NumPy y pandas para el procesamiento de conjuntos de datos en memoria procurando el cumplimiento de las especificaciones de aplicaciones de negocios y ampliación de funcionalidad para evitar la volatilidad asociada a la memoria principal de un proceso. En la **tercera fase** el estudiante construirá soluciones de información que permitan la serialización y la recuperación de elementos de su estado a formatos XML y JSON; así como en sistemas gestores de bases de datos.*



3. Propósito(s):

Estructuras de datos y su procesamiento pretende lograr que el egresado proponga nuevas estructuras de datos o aplique las ya existentes del lenguaje Python para cubrir los requerimientos de almacenamientos de datos e información para resolver las necesidades de procesamiento y almacenamiento de las soluciones tecnológicas, mediante formatos estandarizados y de amplia difusión como lo son XML y JSON, así como la conectividad a sistemas gestores de bases de datos relacionales de alto desempeño.

Esta Unidad de Aprendizaje (UA) se relaciona con la antecedente Programación avanzada ya que se debe conocer el paradigma de programación orientada a objetos y su aplicación para poder aprovechar adecuadamente los conceptos que se abordarán en la presente UA. Así mismo proporciona elementos indispensables al introducir múltiples alternativas de almacenamiento de datos en memoria y almacenamiento secundario que son utilizados como fuente de la visualización de datos que se abordará en la UA sucesora denominada Visualización de datos y desarrollo de escritorio.

Estructuras de datos y su procesamiento contribuye con las competencias generales de UANL, ya que el estudiante desarrollará propuestas individuales y en equipo respetando las diversas ideas de sus compañeros y la retroalimentación del profesor, para así seleccionar y justificar literatura especializada existente en manuales de los fabricantes de software y repositorios de código en línea, las cuales permitirán una innovación en la aplicación o el desarrollo de nuevas estructuras de datos.

Así mismo contribuye con la competencia específica ya que dominará los conceptos y aplicaciones de las estructuras de datos que ofrece Python y los sistemas gestores de base de datos líderes en el mercado para aplicarlos en el proceso de desarrollo de software y la definición y aprovechamiento de almacenes de datos.

4. Competencias del perfil de egreso:

Instrumentales:

1. Aplicar estrategias de aprendizaje autónomo en los diferentes niveles y campos del conocimiento que le permitan la toma de decisiones oportunas y pertinentes en los ámbitos personal, académico y profesional.
2. Utilizar los lenguajes lógico, formal, matemático, icónico, verbal y no verbal de acuerdo con su etapa de vida, para comprender, interpretar y expresar ideas, sentimientos, teorías y corrientes de pensamiento con un enfoque ecuménico.

Personales:

10. Intervenir frente a los retos de la sociedad contemporánea en lo local y global con actitud crítica y compromiso humano, académico y profesional para contribuir a consolidar el bienestar general y el desarrollo sustentable.



Universidad Autónoma de Nuevo León
Facultad de Contaduría Pública y Administración
Programa analítico



Integradoras:

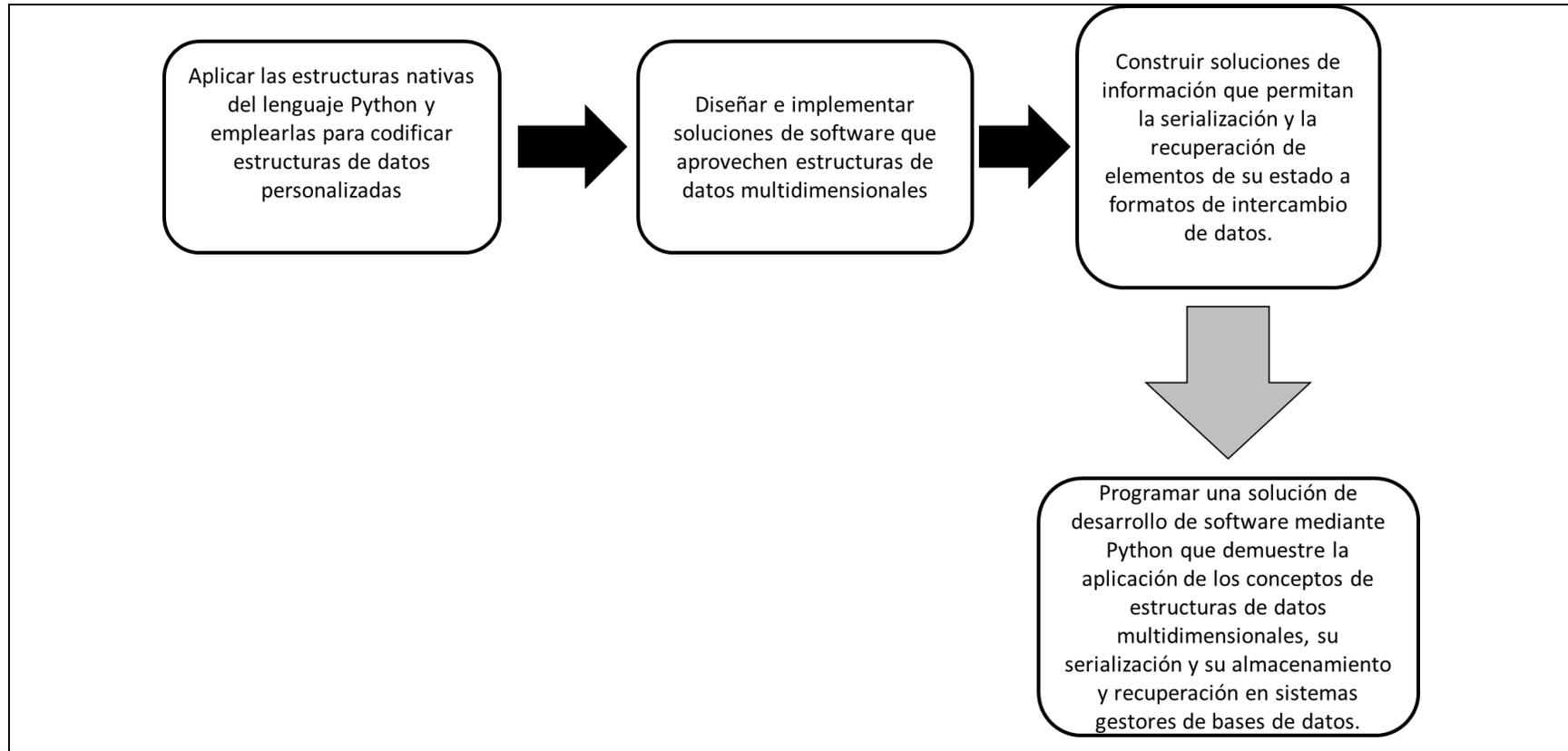
13. Asumir el liderazgo comprometido con las necesidades sociales y profesionales para promover el cambio social pertinente.

Competencias específicas del perfil de egreso a las que contribuye la unidad de aprendizaje:

6. Desarrollar software aplicando lenguajes de programación de alta productividad y amplio campo de aplicación, bajo, estándares, metodologías y mejores prácticas de desarrollo con el propósito de almacenar y procesar datos e información derivados de la operación diaria de la organización



5. Representación gráfica:





6. Estructuración en fases:

Fase 1: Estructuras de datos nativas del lenguaje Python.

Elemento de competencia: Aplicar las estructuras nativas del lenguaje Python tales como Listas, Tuplas, Conjuntos y Diccionarios para almacenar en memoria principal el estado de una aplicación a desarrollar y emplearlas para codificar pilas y colas que permitan el proceso de los conjuntos de datos.

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos.
Repositorio en plataforma Git con soluciones de desarrollo de código Python que demuestren la aplicación de estructuras de datos nativas del lenguaje.	<ul style="list-style-type: none"> • Publicar un repositorio en el que incluirá los trabajos de programación a desarrollar procurando evitar el consumo de papel. • Estructurar el repositorio de acuerdo con las diferentes estructuras de datos a demostrar. • Evidenciar la aplicación de las estructuras de datos nativas del lenguaje. 	<ul style="list-style-type: none"> • El docente expondrá de manera visual y oral basándose en ejemplos las diferentes estructuras de datos nativas a Python, así como sus diferencias entre si y recomendará criterios de selección a tomar en cuenta para este fin. • El estudiante realizará un diagrama de flujo que le permita representar el flujo de decisiones para elegir la estructura de datos nativa más conveniente. • El docente demostrará ejemplos prácticos de programación que apliquen las estructuras de datos nativas del lenguaje. • El estudiante codificará soluciones a casos de estudio que aprovechen las estructuras de datos nativas de Python. 	<ul style="list-style-type: none"> • Ventajas comparativas de las estructuras de datos: Listas, Tuplas, Diccionarios y conjuntos. 	<ul style="list-style-type: none"> • Recursos tecnológicos tales como: <ul style="list-style-type: none"> ○ Equipo de cómputo ○ Proyector ○ Conexión a internet • Software de productividad (Office universitario). • Software de control de versiones (Git). • Intérprete Python 3



Universidad Autónoma de Nuevo León
Facultad de Contaduría Pública y Administración
Programa analítico



Fase 2: Diseñar e implementar soluciones de software que aprovechen estructuras de datos multidimensionales

Elemento de competencia: Diseñar e implementar soluciones de software que aprovechen las estructuras de datos multidimensionales bajo las librerías NumPy y pandas para el procesamiento de conjuntos de datos en memoria para el cumplimiento de especificaciones en aplicaciones de negocios.

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos.
Repositorio en plataforma Git que demuestre el aprovechamiento de estructuras multidimensionales bajo NumPy y Pandas	<ul style="list-style-type: none"> • Publicar un repositorio en el que incluirá los trabajos de programación a desarrollar procurando evitar el consumo de papel. • Estructurar el repositorio de acuerdo con las diferentes estructuras de datos a demostrar. • Demostrar la creación y manipulación de estructuras de datos vectoriales y multidimensionales soportadas bajo NumPy y pandas. 	<ul style="list-style-type: none"> • El docente expondrá de manera visual, oral y mediante ejemplos las funcionalidades que las librerías NumPy y pandas ofrecen referente al manejo de estructuras multidimensionales y su procesamiento. • El estudiante realizará un catálogo de casos de aplicación para las principales funcionalidades de procesamiento de datos en memoria de las librerías NumPy y pandas. 	<ul style="list-style-type: none"> • Principales ventajas de la aplicación de las librerías NumPy y pandas para el procesamiento de datos en memoria. 	<ul style="list-style-type: none"> • Recursos tecnológicos tales como: <ul style="list-style-type: none"> ○ Equipo de cómputo ○ Proyector ○ Conexión a internet • Software de productividad (Office universitario). • Software de control de versiones (Git). Intérprete Python 3



Universidad Autónoma de Nuevo León
Facultad de Contaduría Pública y Administración
Programa analítico



Fase 3: Construir soluciones de información que permitan la serialización y la recuperación de elementos de su estado a formatos de intercambio de datos.

Elemento de competencia: Construir soluciones de información que permitan la serialización y la recuperación de elementos de su estado a formatos XML y JSON; así como en sistemas gestores de bases de datos.

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos.
Repositorio en plataforma Git que evidencie la exportación e importación de datos a formatos estándares de la industria y el almacenamiento del estado en almacenes de datos relacionales.	<ul style="list-style-type: none"> • Publicar un repositorio en el que incluirá los trabajos de programación a desarrollar procurando evitar el consumo de papel. • Estructurar el repositorio de acuerdo con los diferentes formatos para intercambio de datos e interacción con gestores relacionales de base de datos. • Demostrar la exportación/ importación del estado de una aplicación hacia y desde los diferentes formatos de intercambio de datos y sistemas gestores relacionales de bases de datos 	<ul style="list-style-type: none"> • El docente expondrá de manera visual, oral y mediante ejemplos los formatos de intercambio de datos. • El estudiante elaborará un cuadro comparativo entre los formatos de intercambio de datos. • El docente demostrará mediante ejemplos, las mejores prácticas para la exportación de datos mediante los formatos CSV, XML y JSON. • El estudiante resolverá casos de estudio que demuestre la competencia para la 	<ul style="list-style-type: none"> • Importancia de la exportación de datos bajo situaciones de masividad y de compatibilidad interplataforma. • Aprovechamiento de las tareas de DML en sistemas gestores de bases de datos. • Aplicabilidad del manejo de excepciones para procurar la tolerancia a fallos en las soluciones desarrolladas. 	<ul style="list-style-type: none"> • Recursos tecnológicos tales como: <ul style="list-style-type: none"> ○ Equipo de cómputo ○ Proyector ○ Conexión a internet • Software de productividad (Office universitario). • Software de control de versiones (Git). • Intérprete Python 3 • Sistemas gestores de bases de datos (ORACLE, SQL Server, My SQL, etc.)



	implementando un manejo estructurado de excepciones.	serialización de datos en formato CSV, XML y JSON. <ul style="list-style-type: none"> • El docente demostrará las técnicas que permiten expresar las funciones de DML en almacenes de datos. • El estudiante demostrará, mediante la solución de casos de estudio, la conservación y recuperación del estado de una aplicación en un sistema gestor de base de datos dentro de un marco de trabajo que contemple el manejo estructurado de excepciones. 		
--	--	---	--	--

7. Evaluación integral de procesos y productos:

Evidencia	Ponderación
Repositorio en plataforma Git con evidencias de Estructuras de datos nativas	5%
Evaluación de medio término	30%
Repositorio en plataforma Git con estructuras de datos extendidas mediante librerías NumPy y Panda	5%



Repositorio en plataforma Git con evidencias de serialización y almacenamiento de datos en sistemas de gestión de bases de datos	10%
PIA	20%
Evaluación Fin de curso	30%
Total	100%

8. Producto integrador de aprendizaje:

Programar una solución de desarrollo de software mediante Python que demuestre la aplicación de los conceptos de estructuras de datos multidimensionales, su serialización y su almacenamiento y recuperación en sistemas gestores de bases de datos.

9. Fuentes de apoyo y consulta:

Cuevas Álvarez, A. (2017). Python 3. Paracuellos del Jarama, Madrid: Ra-Ma.

Halterman, R.L. (2011) Learn to program in Python. Southern Adventist University

Learnpython.org. (2018). Learn Python - Free Interactive Python Tutorial. [online] Disponible en: <https://www.learnpython.org> [Accesado 26 Oct. 2018].

Lubanovic, B. (2015). Introducing Python. (A. Oram & A. MacDonald, Eds.) (First). O'Reilly.

Saldanha da Gama, F. (2018). Computers & Operations Research Magazine.

Tagliaferri, L. (2013) How to code in Python 3. New York City, New York, USA: DigitalOcean.



Universidad Autónoma de Nuevo León
Facultad de Contaduría Pública y Administración
Programa analítico



Anexo.

Producto integrador de aprendizaje: Reporte de una solución de desarrollo de software mediante Python que demuestre la aplicación de las estructuras de datos multidimensionales, tolerancia a fallos y el almacenamiento permanente del estado de la misma; opcionalmente, publicarla en un repositorio de software colaborativo.		
Instrucciones:	Demostración presencial y en línea de la ejecución del código Python que permita comprobar el cumplimiento de los objetivos establecidos para el proyecto abordado.	
Valor:	20	
Criterios de evaluación:	<ul style="list-style-type: none"> a) Introducción en inglés y en español (Presentar caso y antecedentes) b) Propósito (Especificación de los beneficios esperados) (ANECA 2.5) c) Identificación y cumplimiento de los requerimientos funcionales (Listado y argumentación) (ANECA 2.5) d) Diagramas de flujo que modelen los principales procesos de la solución desarrollada (ANECA 2.3) e) Demostrar la serialización bidireccional y almacenamiento no volátil del estado de la solución respaldado por el manejo de excepciones (Identificación y explicación del código que lo implementa) f) Índice congruente con el contenido g) Conclusiones individuales y general (Concisas y pertinentes) h) Competencia Comunicativa (Ortografía, Redacción y legibilidad) i) Fuentes y referencias j) Valores UANL 	<ul style="list-style-type: none"> a) 5% b) 10% c) 15% d) 5% e) 46% f) 3% g) 3% h) 5% i) 3% j) 5%
Modalidad:	Presencial	
Subresultados ANECA	<p>2.3 Seleccionar y utilizar los correspondientes métodos analíticos, de simulación y de modelización.</p> <p>2.5 Analizar la medida en la que un determinado sistema informático cumple con los criterios definidos para su uso actual y desarrollo futuro.</p>	