|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Gestión de analítica de datos |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | Implementar la solución de software de acuerdo con los requisitos de operación y modelos de referencia. | RESULTADOS DE APRENDIZAJE | 220501097-01 Desplegar la solución informática, de acuerdo a criterio técnico y marcos de referencia. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | CF19 |
| NOMBRE DEL COMPONENTE FORMATIVO | Despliegue de solución informática |
| BREVE DESCRIPCIÓN | El despliegue de una solución informática es una de las fases más importantes dentro del proceso de la analítica de datos. Es la fase visible para el usuario o cliente del sistema, por lo tanto, las aplicaciones o modelos implementados deben garantizar su usabilidad, disponibilidad y entrega de respuestas oportunas para las problemáticas que fueron diseñadas. |
| PALABRAS CLAVE | Despliegue, Heroku, Docker, API, AWS. |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | Ciencias naturales, aplicadas y relacionadas. |
| IDIOMA | Español |

**Tabla de contenidos**

**Introducció**n

1 **Introducción a la implementación de modelos de *Machine Learning***

1.1 Implementación de Pipelines

1.2 Optimización de la implementación de modelos

1.3 Arquitectura del sistema de *Machine Learning*

2 **Implementar modelos a través de API REST**

2.1 Ejecución de la API localmente

2.2 Heroku (Plataforma como servicio PaaS)

2.3 Despliegue de la API con contenedores

2.4 Docker

**3 Implementación en IAAS (AWS ECS)**

3.1 Introducción a AWS

3.2 Costos y precauciones de AWS

3.3 Manejo de AWS

3.4 Despliegue en ECS

# Introducción

|  |
| --- |
| Cuadro de texto |
| Estimado aprendiz, reciba una cordial bienvenida a este recurso de aprendizaje orientado a la ciencia de datos. Durante su desarrollo se abordarán los ejes temáticos: introducción a la implementación de modelos de *machine learning*, implementar modelos a través de API REST e implementación en IAAS (aws ecs). Al finalizar el componente formativo, estará en la capacidad de desplegar la solución informática de acuerdo a criterio técnico y marcos de referencia. Antes de empezar, mire el siguiente video para identificar el contexto de aprendizaje. ¡Muchos éxitos, que tenga un aprendizaje significativo y memorable! |

**Guion de video introductorio**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo de recurso** | Video motion | | | |
| **NOTA** | **La totalidad del texto locutado para el video no debe superar las 500 palabras aproximadamente** | | | |
| **Título** | **Despliegue de solución informática** | | | |
| **Escena** | **Imagen** | **Sonido** | **Narración** | **Texto** |
| 1 | Imagen: 228131\_i1901 <https://www.pngwing.com/es/free-png-nwrwm> | NA | Uno de los retos más desafiantes en la ciencia de datos es el lanzamiento del modelo que se encuentra entrenado a producción para que cualquier usuario u organización pueda usarlo fácilmente. | Lanzamiento del modelo |
| **2** | Imagen: 228131\_i1902  <https://www.pngwing.com/es/free-png-blzlu> | NA | Los analistas de datos deben decidir qué recursos computacionales se necesitan para configurar la infraestructura adecuada que pueda replicar el modelo de los ambientes de desarrollo en un entorno de producción logrando resultados similares | Recursos computacionales |
| **3** | Imagen: 228131\_i1903  <https://www.pngwing.com/es/free-png-zuaww> | NA | El despliegue de una aplicación o modelo de *machine learning* se inicia de forma local o en entornos de prueba | Despliegue local |
| **4** | Imagen: 228131\_i1904  <https://www.pngwing.com/es/free-png-vomss> | NA | donde el analista de datos realiza las verificaciones que garantizan que se dé solución a las necesidades por las cuales se construyó el modelo. | Realiza las verificaciones |
| **5** | Imagen: 228131\_i1905  <https://www.pngwing.com/es/free-png-dzhxn> | NA | La mayor parte de los recursos disponibles se consumen en el proceso de creación del modelo con el objetivo de alcanzar los resultados deseados | Creación del modelo |
| **6** | Imagen: 228131\_i1906  <https://www.pngwing.com/es/free-png-nerhb> | NA | por lo que se asignan muchas horas de trabajo y energía en la etapa de preprocesamiento y entrenamiento del modelo. | Entrenamiento del **modelo** |
| **7** | Imagen: 228131\_i1907  <https://www.pngwing.com/es/free-png-adfcq> | NA | Cuando los modelos han sido verificados y generan las respectivas predicciones, el analista datos debe decidir qué plataforma de despliegue se usará para la producción. | Decidir la plataforma de despliegue |
| **8** | Imagen: 228131\_i1908  <https://www.pngwing.com/es/free-png-prnmw> | NA | Esta plataforma, por lo general, está ubicada en la nube y debe cumplir con criterios de calidad, seguridad y alta disponibilidad. | Ubicada en la nube |
| **9** | Imagen: 228131\_i1909  <https://www.pngwing.com/es/free-png-hyhlt> | NA | En el mercado existen varios proveedores de tecnología que ofrecen servicios y herramientas para fortalecer la implementación de soluciones informáticas. | Hay varios proveedores |
| **10** | Imagen: 228131\_i1910  <https://www.pngwing.com/es/free-png-yvcu> | NA | Cada uno de ellos permite la integración de los desarrollos locales con las plataformas en la nube y facilita la administración y visualización, en tiempo real, de las características y funcionalidades de las aplicaciones desplegadas. | Facilita la administración y visualización |
| **11** | Imagen: 228131\_i1911  <https://www.pngwing.com/es/free-png-stnwc> | NA | Es importante que antes de implementar una aplicación o modelo se analice detalladamente con el usuario o cliente el costo de un error en la predicción, | Costo que genera un error |
| **12** | Imagen: 228131\_i1912  <https://www.pngwing.com/es/free-png-agntq> | NA | ya que es posible que el impacto sea mayor si la predicción es errada para un no que para un sí. | El impacto sea mucho mayor |
| **13** | Imagen: 228131\_i1913  <https://www.pngwing.com/es/free-png-vttcu> | NA | En la etapa final de despliegue se deben priorizar las capacitaciones sobre el funcionamiento del modelo al usuario final | Capacitaciones al usuario final |
| **14** | Imagen: 228131\_i1914  <https://www.pngwing.com/es/free-png-djqhl> | NA | para que este realice un uso adecuado de la aplicación y se maximice el éxito en su implementación | Maximizar casos de éxito |
| **15** | Imagen: 228131\_i1915  <https://www.pngwing.com/es/free-png-twrzo> | NA | Es fundamental realizar simulaciones previas a la puesta en operación de la solución informática para detectar errores en la lógica de programación | Implementación de simulaciones |
| **16** | Imagen: 228131\_i1916  <https://www.pngwing.com/es/free-png-zkjhd> | NA | Y posibles fallas en la arquitectura que pongan en riesgo el éxito del modelo en su fase de producción. | Coloque en riesgo el éxito |
| **17** | Imagen: 228131\_i1917  <https://www.pngwing.com/es/free-png-dyyyx> | NA | La generación de reportes es un insumo valioso en la implementación de una aplicación o modelo de analítica de datos | Generación de reportes |
| **18** | Imagen: 228131\_i1918  <https://www.pngwing.com/es/free-png-vpujx> | NA | porque permite a los usuarios del sistema visualizar los datos de forma gráfica para poder interpretarlos más fácilmente. | Permite interpretar los datos |
| **19** | Imagen: 228131\_i1919  <https://www.pngwing.com/es/free-png-zilyo> | NA | En este componente formativo aprenderá a optimizar modelos de *machine learning* para su posterior despliegue a producción | Optimizar modelos |
| **20** | Imagen: 228131\_i1920  <https://www.pngwing.com/es/free-png-shrnm> | NA | así mismo conocerá las distintas opciones y herramientas que existen para la implementación y despliegue de las soluciones informáticas en entornos locales y en la nube. | Implementación y despliegue |
| **21** | Imagen: 228131\_i1921  <https://www.pngwing.com/es/free-png-nqtny> | NA | Además, comprenderá las ventajas que ofrece uno de los proveedores más grandes y populares en el despliegue de las aplicaciones informáticas. | Proveedor más grande y popular |
| **Nombre del archivo** | 228131\_v01 | | |  |

**Desarrollo de contenido**

# Introducción a la implementación de modelos de *Machine Learning*

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Slider Presentación | |
| **Introducción** | El proceso de implementación de modelos de *Machine Learning* consiste en pasar un modelo previamente construido hacia un ambiente donde estará disponible para ser accedido por los usuarios o clientes. Este ambiente, por lo general, está alojado en la nube y a través de internet proporciona disponibilidad para que se pueda usar en tiempo real. | |
| La implementación o despliegue de modelos de *machine learning* ofrece a las empresas ventajas competitivas en el mercado y contexto actual, ya que, permite integrarlos con los demás módulos del ecosistema de la organización, generar valor agregado al negocio y maximizar la operación comercial. | | **Figura 1** *Ecosistema*    Imagen: 228131\_i1922 |
| Los modelos de *Machine Learning* se pueden aplicar en todas las industrias, por ejemplo, en un sector como el comercio electrónico, donde los usuarios realizan compras a través de un sitio web, se implementan sugerencias o recomendaciones sobre los productos de acuerdo al perfil del cliente creado previamente para maximizar la probabilidad de compra. | | Imagen: 228131\_i1923 |
| Las implementaciones actuales necesitan que el flujo y procesamiento de la información sea sincrónico, es decir que la información debe ser procesada al instante. Además, deben soportar el registro y consulta de grandes volúmenes de datos desde diferentes fuentes y tipos. | | Imagen: 228131\_i1924 |

## Implementación de Pipelines

|  |  |
| --- | --- |
| Tipo de recurso | Cajón de texto de color |
| Un pipeline de datos es un conjunto de fases y herramientas tecnológicas que se integran para realizar los procesos de transformación de los datos desde el inicio hasta su almacenamiento persistente. La creación de Pipelines es un componente central de la ciencia de datos que permite crear aplicaciones para recopilar datos de millones de usuarios y procesar los resultados casi en tiempo real. Base de datos de documentación en línea y concepto de sistema de gestión de documentos.  Empresaria trabajando en laptop con pantalla virtual. Automatización de procesos para administrar archivos de manera eficiente.  Así mismo, los pipelines se entienden como una composición de tareas, donde cada una toma un conjunto de entradas y produce un conjunto de salidas. Estas canalizaciones combinan las tareas en formas especificadas por el analista de datos que las crea (Cedeno, 2020).  Imagen: 228131\_i1925 | |

|  |  |
| --- | --- |
| **Tipo de recurso** | Acordeón tipo 1 |
| **Introducción** | Generalmente, los pipelines canalizan los datos hacia repositorios o lagos de datos que se encuentran en la nube, como Hadoop, S3 de AWS o bases de datos relacionales como Redshift.  Los pipelines tienen las siguientes propiedades: |
| Imagen: 228131\_i1926 | |
| **Baja latencia**  Un pipeline permite consultas de datos a segundos de la fase final del proceso. Esto permite, a los científicos de datos, la creación de productos que se actualicen de forma inmediata. | |
| **Escalabilidad**  Un pipeline se puede utilizar tanto para procesar un centenar de datos como para miles o millones de ellos. Los sistemas de alto rendimiento no solo deben tener la capacidad de almacenar datos, sino también ofrecer el acceso para consultas de la totalidad de estos. | |
| **Control de versiones**  Un pipeline podrá realizar cambios de versión sin interrumpir el proceso de los datos o generar pérdida de estos. | |
| **Monitoreo**  Los pipelines deben generar alertas cuando en una fase no se reciben datos o eventos, esto con el fin de que el analista de datos examine la anomalía y realice los correctivos pertinentes. | |
| **Testing**  Los pipelines permiten las pruebas de datos cuando ingresan a una de sus fases, aun cuando no terminan en un base o lago de datos. | |

|  |  |
| --- | --- |
| **Tipo de recurso** | Rutas / Pasos. Verticales 1 |
| **Introducción** | Uno de los estándares más comunes para el almacenamiento de datos masivos en la industria es el *data lake* o lago de datos, el cual permite almacenar datos semiestructurados en una base de datos distribuida, ejecutar procesos ETL para extraer los datos más relevantes y almacenarlos en las bases de datos de análisis. Se pueden usar diferentes herramientas para la base de datos distribuida: como Hadoop, Cosmos o S3. |
| **Figura 2** *Estructura de pipeline con lago de datos*    Imagen: 228131\_i1927 | |
| **Botón 1** | Los lagos de datos proporcionan a los analistas la capacidad de procesar información desde distintos volúmenes de datos con altos niveles de tolerancia a fallas. Sin embargo, presentan complejidad en el acceso a los datos en comparación con las bases datos tradicionales, debido a la falta de herramientas o políticas de acceso. |
| **Botón 2** | Los pipelines son ampliamente usados en la automatización de flujos de trabajo para ahorrar tiempo y aumentar la eficiencia. El conocimiento de los procesos a automatizar es importante en el diseño del pipeline para que se pueda repetir en varios servicios y configurarse una sola vez. |
| **Botón 3** | En el campo del *Data Science,* los pipelines hacen que el preprocesamiento de los datos se realice muy rápido. Además, el pipeline se puede reutilizar en las fases de pruebas o testeo donde los datos aún no se han transformado o con nuevos datos que ingresen al modelo. |

|  |  |
| --- | --- |
| **Tipo de recurso** | Acordeón tipo 2 |
| **Introducción** | A continuación, se muestra un ejemplo de la creación de un pipeline usando: el lenguaje de programación Python, la librería *Scikit Learn* y la herramienta Google Colab. |
| Imagen: 228131\_i1928 | |
| **Paso 1: importar librerías**  # Se importan las librerías esenciales de pandas y numpy  import pandas as pd  import numpy as np  # Se importa Pipeline  from sklearn.pipeline import Pipeline  # Se importan los requisitos para el funcionamiento del modelo  from sklearn.preprocessing import StandardScaler  from sklearn.linear\_model import LinearRegression  from sklearn.model\_selection import train\_test\_split | |
| **Paso 2: crear conjunto de datos**  Ahora crearemos un conjunto de datos para el ejemplo:  # Creación aleatoria de un Dataframe  Caracteristica1 = np.random.randint(10,99, 999)  Caracteristica2 = np.random.choice(np.random.randint(15,200), 999)  a = np.random.randn(999)  y1 = 0.195  y2 = 0.377  dataframe = pd.DataFrame({ 'grupo': np.random.choice([10, 20, 30], 999),  'Caracteristica1': Caracteristica1 ,  ' Caracteristica2': Caracteristica2,  'Tipo': np.random.choice([11, 12, 21, 22, 3], 999),  'Resultado': ((Caracteristica1\*y1) + (Caracteristica2\*y2) + a) }) | |
| **Paso 3: aplicar modelo de regresión**  El conjunto de datos creado tiene la siguiente estructura. A estos datos se les aplicará un modelo de regresión con el fin de predecir la variable resultado.  Figura Dataframe  Imagen que contiene Tabla  Descripción generada automáticamente  Imagen: 228131\_i1929 | |
| **Paso 4: crear lista de tuplas**  Para crear pipelines es indispensable conocer previamente los pasos que conforman el proceso, la notación es “( nombre\_del\_paso', Instancia() )”, donde el primer parámetro es el nombre del paso y el segundo, su respectiva instancia.  Luego de conocer los pasos del pipeline se creará una lista de tuplas:  # Crear los pasos a realizar  pasos = [('Estandar', StandardScaler()),           ('Regresion\_Lineal', LinearRegression()) ] | |
| **Paso 5: crear objeto**  Seguidamente, se crea el objeto para el pipeline o tubería.  tuberia = Pipeline(pasos) | |
| **Paso 6: crear *dataframe* de entrenamiento**  La siguiente fase no hace parte de la creación del pipeline, pero será necesaria para la articulación e implantación del modelo  # Se crea un nuevo *dataframe* para el entrenamiento del modelo  df\_train = dataframe  #Se separa la columna que contiene la variable objetivo "Resultado"  X = np.array(df\_train.drop(['Resultado'],1))  y = np.array(df\_train['Resultado'])  #Se dividen los datos en entrenamiento y prueba  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 10)  #Se usa 20% de datos para prueba y una semilla de 10. | |
| **Paso 7: uso del pipeline**  Finalmente, se hace uso del pipeline en los procesos de entrenamiento, medición de métricas y predicción:  # Pipeline para el entrenamiento  tuberia.fit(X,y)  # Pipeline para el Score  tuberia.score(X, y)  # Pipeline para la predicción  tuberia.predict(X\_test) | |

|  |
| --- |
| Cuadro de texto |
| En el ejemplo anterior se muestra la implementación de un pipeline para realizar el proceso de entrenamiento del modelo y la predicción de la variable resultado con el algoritmo de regresión lineal. |

|  |
| --- |
| Cuadro de texto |
| Para profundizar en el uso de los pipelines con Scikit Learn consulte el artículo: *‘****How to Use Sklearn Pipelines For Ridiculously Neat Code’***, disponible en <https://towardsdatascience.com/how-to-use-sklearn-pipelines-for-ridiculously-neat-code-a61ab66ca90d> |

## Optimización de la implementación de modelos

|  |  |
| --- | --- |
| Tipo de recurso | Cajón de texto de color |
| Generalmente, las organizaciones cuentan con datos sin procesar recopilados y almacenados en bases de datos. Estos datos no son adecuados para entrenar modelos de *Machine Learning* o como insumo a modelos ya existentes para generar una predicción. El ingeniero de datos, por su parte, debe realizar una serie de transformaciones antes de que los algoritmos de *Machine Learning* puedan utilizar estas variables. A estas transformaciones se les conoce como ingeniería de funciones.  Imagen: 228131\_i19xx | |

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | | Pestañas o tabs Verticales |
| **Introducción** | | La ingeniería de funciones utiliza el conocimiento que el experto del dominio tiene de los datos para crear métodos que realicen imputación de datos faltantes, codificación de variables categóricas, transformación de variables numéricas y creación de nuevas características, entre otras. Estos métodos deben devolver un conjunto de datos idóneo para el entrenamiento de los modelos de aprendizaje automático.  Los científicos de datos dedican hasta el 60% del tiempo en tareas relacionadas con la ingeniería de funciones o características. Este tiempo se justifica por las siguientes razones: |
| Imagen: 228131\_i1930 | | |
| **1** | Librerías populares de *Machine Learning* como Scikit-Learn no admiten campos faltantes o datos categóricos en sus entradas, por lo tanto, esos campos deben ser imputados con algún valor que los represente como la media o la moda. | |
| **2** | Algunos algoritmos como la regresión lineal, las redes neuronales y el K vecino más cercano son sensibles a la escala variable, de modo que todos los valores de una variable deben tener la misma unidad o métrica, por ejemplo, en un campo que contiene datos de tiempo, todos los registros deben entenderse ya sea por días, semanas, meses o años, pero no deben contener diferentes escalas. | |
| **3** | Algunos algoritmos como la regresión lineal son muy sensibles a valores atípicos, por ejemplo, si tenemos una base de datos con información sobre el costo de las casas en una región del país y uno de sus registros tiene un valor excesivamente bajo, este valor se cataloga como atípico y debe tratarse, ya sea con una imputación o una eliminación. | |
| **4** | Se pueden obtener muchos datos útiles de los datos en crudo, por ejemplo, fechas, series temporales, transacciones entre otros. | |

|  |  |
| --- | --- |
| **Tipo de recurso** | Acordeón tipo 1 |
| **Introducción** | En la creación de nuevas características para la optimización de los modelos de *machine learning* se destacan los siguientes procesos de la ingeniería de funciones: |
| Imagen: 228131\_i1931 | |
| **Imputación de datos faltantes**  Este proceso consiste en el reemplazo de valores faltantes de la base de datos por números. Estos números, por lo general, son estimaciones estadísticas. A continuación, se muestra un ejemplo de esta técnica empleando la clase *SimpleImputer* de librería Scikit-Learn:  # Se importan las librerias requeridas para el proceso de imputación  from sklearn.impute import SimpleImputer  import numpy as np  # Se crea una matriz de ejemplo con datos que contiene valores vacios  matriz = [[3, 8], [np.nan, 6], [1, 5]]  print(matriz)  #Salida: [[3, 8], [nan, 6], [1, 5]]  # se crea el objeto de imputación con la media estadística  imputacion = SimpleImputer(missing\_values=np.nan, strategy='mean')  # Se imputan los valores vacíos que encuentra en la matriz  imputacion = imputacion.fit(matriz)  # Se transforman los datos de la matriz  matriz = imputacion.transform(matriz)  # se muestra la matriz transformada  print(matriz)  #Salida:  #[[3. 8.]  # [2. 6.]  # [1. 5.]] | |
| **Discretización**  Este proceso ordena los valores de una variable en intervalos con el fin de obtener un número limitado de estados. Generalmente los analistas de datos crean intervalos de igual ancho o frecuencia para clasificar los valores de las variables. Por ejemplo, cuando se requiere ordenar el campo de edades de una base de datos en los estados de: niño, adolescente, adulto y anciano. Modelos como los *árboles de decisión* y *naive bayes* tienen mejor rendimiento trabajando con valores discretos. A continuación, se muestra un ejemplo de esta técnica usando la librería pandas de Python  # En el ejemplo siguiente se discretiza la variable "edad"  # en un conjunto de datos, utilizando el método de intervalos iguales.  # Se utilizarán tres intervalos:  # 12-21 años  # 22-32 años  # 33-75 años  # Se importa la librería pandas  import pandas as pd  # Se crea un set de datos con valores aleatorios  datos = pd.DataFrame({'edad': [19, 50, 28, 21, 29, 33, 24, 45, 45, 52,                                 51, 52, 28, 53, 55, 33, 64, 39, 22]})  # Se discretiza la variable "edad"  datos["edad\_discreta"] = pd.cut(datos["edad"], bins=[12, 21, 33, 75],                                  labels=["12-21", "22-32", "33-75"])  # Se muestran los 5 primeros valores de forma discreta  print(datos["edad\_discreta"].head(5))  # Salida:  #0 12-21  #1 33-75  #2 22-32  #3 12-21  #4 22-32 | |
| **Codificación de variables categóricas:**  Las variables categóricas son aquellas que tienen nombres de categorías y no números como valores, por ejemplo, los valores de la variable “*estado civil”* pueden ser: soltero, divorciado, casado y otros. Es habitual que los analistas de datos agrupen los valores poco comunes o raros en la categoría llamada *otros*. A continuación, se muestra un ejemplo de esta técnica empleando la clase *LabelEncoder* de librería Scikit-Learn:  # Se importan las librerías  import pandas as pd  from sklearn.preprocessing import LabelEncoder  # Se crea un dataframe con los nombres de las categorías  dataframe = pd.DataFrame({"variable": ['soltero','casado','soltero','dicorciado','casado']})  # Se crea el objeto de codificación  codificacion = LabelEncoder()  # Se codifican las variables  dataframe["variable"] = codificacion.fit\_transform(dataframe["variable"])  # Se muestra el dataframe codificado  print(dataframe)  # Salida:  # variable  #0 2  #1 0  #2 2  #3 1  #4 0 | |

|  |
| --- |
| Cuadro de texto |
| Para una mejor comprensión del tema *Machine Learning*, se le invita a ver el siguiente video:  **VIDEO TEMÁTICO: 228131\_v02 – video clase**  VIDEO CLASE GRABADA POR EL EXPERTO - SE ENCUENTRA EN LA CARPETA CF19\_228131 – VIDEOS |

|  |
| --- |
| Cuadro de texto |
| **Descarga de archivos**  Para realizar los ejercicios explicados en la videoclase, descargue los archivos que se usaran en la práctica:  Anexos:   1. Datos\_A.xlsx 2. Datos\_B.xlsx 3. Optimizacion\_de\_Dataset.ipynb   NOTA: SE ENCUENTRAN DENTRO DE LA CARPETA CF19\_228131\_Anexos videoclase como **MATERIAL DESCARGABLE.ZIP** |

## Arquitectura del sistema de Machine Learning

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | | Pestañas o tabs Verticales |
| **Introducción** | | Los modelos de *Machine Learning* se crean utilizando distintos lenguajes de programación como Java, Python, R, Julia, entre otros. A su vez, cada lenguaje cuenta con librerías de apoyo para su construcción, por ejemplo, Python tiene scikit-learn, TensorFlow, PyTorch, etc. Teniendo en cuenta los diversos lenguajes y librerías con los que se pueden crear los modelos de *machine learning*, es necesario exportarlos en un estándar que funcione en cualquier plataforma. |
| **Figura 3** *Exportación de Modelos Multiplataforma*    Imagen: 228131\_i1932 | | |
| **Estándares** | Los estándares para exportación de modelos de *machine learning* son la solución para su portabilidad y disponibilidad, independientemente de la plataforma donde se desplieguen. Entre los estándares más populares se encuentran:   * **PMML:** El estándar PMML (*Predictive Model Markup Language*) es un lenguaje de marcado que tiene como base a XML. Fue desarrollado por el Grupo de Minería de Datos (DMG) con el fin de soportar modelos estadísticos y de minería de datos. * **ONNX:** El estándar ONNX(*Open Neural Network Exchange*) es idóneo para los modelos de *Deep Learning*, ya que produce un diagrama de la red que se guarda en un archivo binario y puede ser utilizado en distintas plataformas. | |
| **Diseño del modelo** | En los ambientes de producción no importa la plataforma en que se diseñó y construyó un modelo de *Machine Learning*, ya que se debe garantizar que los componentes desarrollados puedan consumirse desde distintas plataformas, por ejemplo, que un modelo diseñado en Python pueda ser utilizado desde una aplicación con sistema operativo Android. | |
| **Ambientes de producción** | La ciencia de datos generalmente se realiza sobre Python y sus librerías, sin embargo, en los ambientes de producción su uso no es recomendable, debido a que su baja escalabilidad lo limita para procesar grandes cantidades de datos. El entrenamiento de modelos con petabytes de información requiere de plataformas de *Big Data* como Hadoop, Spark, Flink, etc, las cuales son preferidas por los sistemas de producción de las organizaciones. | |
| **Arquitectura** | Los sistemas de *machine learning* tienen los siguientes desafíos en los diseños de su arquitectura:   * **Python** es ideal para la implementación de modelos de alto rendimiento, sin embargo, requiere aumentar su capacidad de procesamiento y escalabilidad para el entrenamiento de modelos con grandes volúmenes de datos.      * Las plataformas como Hadoop, Spark o Flink, aunque solucionan el inconveniente de la escalabilidad, no son adecuadas para integraciones sincrónicas donde el cliente requiere de altos rendimientos en los modelos. | |

# Implementar modelos a través de API REST

|  |  |
| --- | --- |
| **Tipo de recurso** | Acordeón tipo 2 |
| **Introducción** | En estes apartado hablaremos sobre uno de los conceptos más mencionados en el mundo del desarrollo de software, hablamos de API REST. |
| Imagen general que ilustre el tema    Imagen: 228131\_i1933 | |
| **API**  Es el acrónimo de *Interfaz de Programación de Aplicaciones*. Básicamente, es un intermediario de software que permite que dos aplicaciones se comuniquen entre sí. Por ejemplo, cuando se envía un mensaje de texto a través de un servicio de mensajería se está haciendo uso de las API. | |
| **REST**  Es el acrónimo de *Transferencia de Estado de Representación*. Es una interfaz que permite la conexión entre varios sistemas que utilizan el protocolo HTTP y determina cómo los usuarios visualizan la API. REST emplea los métodos de HTTP como GET, PUT, POST y DELETE para crear, modificar y borrar recursos de las bases de datos (López, 2019). | |
| **API REST**  Transfiere al usuario el estado de un recurso solicitado, lo cual es de gran utilidad en la analítica de datos, porque permite solicitar a los modelos de *Machine Learning* sus respectivas predicciones. Para ello, el servidor pasará las predicciones a un usuario que puede ser una aplicación web, una APP móvil o cualquier plataforma que soporte HTTP.  Una API REST brinda las siguientes ventajas en el despliegue de una solución de *Machine Learning*:   * Permite desplegar predicciones sincrónicas que sirven para aumentar el número de usuarios. * Permite combinar múltiples modelos dependiendo de los requerimientos finales de las API. * Permite escalar la solución con el aumento de instancias a través de un balanceador | |

## Ejecución de la API localmente

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Pestañas o tabs horizontales | |
| **Introducción** | Los analistas de datos crean modelos de inteligencia artificial que requieren de aplicaciones web para que los usuarios puedan usarlos. Luego de la creación y entrenamiento de un modelo de aprendizaje automático, este se debe preparar para enviar a producción, sin embargo, es apropiado que antes pase por los ambientes de desarrollo y prueba. En estos ambientes se realizan despliegues a través de APIS locales y, por lo tanto, solo se podrá acceder a ellos desde la misma red. Para profundizar en este tipo de despliegues, se mostrará la puesta en producción local de un modelo de *machine learning*, utilizando el Frameworks Flash del lenguaje de programación Python. | |
| **Instalar**  **Frameworks** | Se recomienda instalar Python a través de la herramienta *Anaconda*, ya que, cuenta con librerías preinstaladas; sin embargo, el Frameworks Flash debe ser instalado de forma independiente. Una opción para instalar Flash en un entorno local es a través de una consola de comandos o Shell, donde se deben ejecutar las siguientes líneas de comandos:  pip install Flask  pip install Flask-RESTful | Instalación del concepto del indicador de carga de barra en espera  Imagen: 228131\_i1934 |
| **Crear modelo** | Una vez instaladas las herramientas, se inicia con la creación del modelo a desplegar. En este ejemplo se trabajará con un *dataset* o conjunto de datos de scikit-learn llamado ***Iris***, que contiene información de una planta con tres clases de flores: virginica, setosa y versicolor. Cada clase tiene propiedades distintas que permiten clasificarla y predecir, a partir de estas variables, qué tipo de clase es. | Máquina de base de datos compartida ilustración de red híbrida, concepto de transmisión de servidor de bases de datos centralizadas, diseño de iconos vectores de almacenamiento de datos y acceso, computación en nube y servicios de alojamiento de Internet Símbolo  Imagen: 228131\_i1935 |
| **Algoritmo de**  **entrenamiento** | Para realizar el entrenamiento se utilizará el algoritmo de árboles de decisión. Luego de entrenado el modelo, se guardará su estado con la librería **joblib** y de este modo, en el proceso de despliegue, solo será necesario llamar al modelo e ingresar los nuevos datos a predecir sin necesidad de entrenar nuevamente. Se inicia con la creación de un archivo de nombre ***modelo.py*** que contiene el siguiente código: | # Se importan las librerías necesarias  from sklearn import datasets  from sklearn.model\_selection import train\_test\_split  from sklearn.tree import DecisionTreeClassifier  import joblib  from sklearn.metrics import accuracy\_score  # Se crea una función para el entrenamiento del modelo  def train\_model():  # Se cargan los datos del dataset Iris a un dataframe      iris\_df = datasets.load\_iris()    # Se separa la variable objetivo del dataset      data = iris\_df.data      objetivo = iris\_df.target    # Se dividen los datos en entrenamiento y pruebas (80% y 20% respectivamente)      X\_train, X\_test, y\_train, y\_test = train\_test\_split(data, objetivo, test\_size=0.20)  # Se entrena el modelo      model = DecisionTreeClassifier().fit(X\_train, y\_train)    # Se obtiene la predicción      prediccion = model.predict(X\_test)    # Se obtiene la precisión      accuracy = accuracy\_score(y\_test, prediccion)    # Se guarda el modelo      joblib.dump( model, 'iris-model.model')      print('Entrenamiento del modelo finalizado\n La exactitud es: {}'.format(accuracy)) |
| **Lanzar modelo** | Con el modelo entrenado y almacenado se crea un nuevo archivo de nombre **app.py,** que tendrá las instrucciones necesarias para lanzar el modelo a producción: | # Se importan las librerías requeridas  from flask import Flask, jsonify, request  # importación del modelo  from model.Train import train\_model  from flask\_restful import Api, Resource  import os  import joblib  # Se crea una instancia de Flask  app = Flask(\_\_name\_\_)  api = Api(app)  # Se verifica que el modelo esta entrenado  if not os.path.isfile('modelo.model'):      train\_model()  modelo = joblib.load('modelo.model')  # Se declara una clase para hacer predicciones  class HacerPrediccion(Resource):      @staticmethod      def post():          datos\_publicados = request.get\_json()          Longitud\_sepalo = datos\_publicados['sepal\_length']          ancho\_sepalo = datos\_publicados['sepal\_width']          Longitud\_petado = datos\_publicados['petal\_length']          ancho\_petalo = datos\_publicados['petal\_width']  # Se realiza la predicción          prediccion = modelo.predict([[Longitud\_sepalo, ancho\_sepalo, Longitud\_petado, ancho\_petalo]])[0]          if prediccion == 0:              predicted\_class = 'Es una Iris setosa'          elif prediccion == 1:              predicted\_class = ' Es una Iris versicolor'          else:              predicted\_class = ' Es una Iris virginica'          return jsonify({              'Prediccion': predicted\_class          })  api.add\_resource(HacerPrediccion, '/prediccion')  if \_\_name\_\_ == '\_\_main\_\_':      app.run( debug = True ) |
| **Ejecutar el modelo** | Finalmente se puede lanzar el modelo ejecutando ***app.py*** desde la consola y realizar predicciones desde un navegador web a través de <http://127.0.0.1:5000/>. En este punto no se tiene una interfaz gráfica que permita ingresar las variables requeridas por el modelo para realizar la predicción, por tanto, se puede utilizar una herramienta como Selenium o Postman para enviar datos a través del método POST y JSON. | Trasero detrás de la vista retrato de su simpático tipo geek enfocado tecleando css analizando el edificio de seguridad del ciberespacio en moderno interior estilo industrial pared de hormigón estación de trabajo  Imagen: 228131\_i1936 |

## Heroku (Plataforma como servicio PaaS)

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | | Pestañas o tabs Verticales |
| **Introducción** | | Heroku es una plataforma como servicio (**PaaS**) que se ejecuta en la nube con capacidad para soportar diferentes lenguajes de programación. Al igual que todas las PaaS permite la manipulación de servidores y sus respectivas configuraciones. Esta plataforma se ha convertido en una de las más populares gracias a su enfoque en el despliegue de aplicaciones.  Heroku despliega aplicaciones en contenedores virtuales que se ejecutan en un entorno seguro. Estos contenedores son llamados Dynos. Heroku permite al desarrollador escalar la aplicación de forma instantánea, simplemente aumentando el número de Dynos. (Danielsson et al., 2021).  Las características principales de Heroku son: |
| Imagen: 228131\_i1937 | | |
| **Multifuncional**: | **Multifuncional**:  Esta plataforma permite la ejecución de lenguajes de programación como Java, Python, Ruby, Node, Go, PHP, Scala, entre otros. | |
| **Diversidad**: | **Diversidad**:  Con Heroku se pueden ejecutar aplicaciones a través de contenedores y más de 200 complementos. | |
| **Seguridad**: | **Seguridad**:  Esta plataforma ofrece mecanismos de seguridad como certificados SSL, autenticación de múltiples factores, monitoreo de amenazas, actualizaciones de seguridad constante, mitigación de ataques de denegación de servicios o DDoS, copias de seguridad entre otros. | |
| **Monitoreo de rendimiento:** | **Monitoreo de rendimiento:**  La plataforma permite supervisar el rendimiento, la memoria, los tiempos de respuesta, la carga de la CPU y las fallas del sistema. | |
| **Escalabilidad**: | **Escalabilidad**:  Heroku permite aumentar y disminuir automáticamente la infraestructura contratada, lo cual garantiza que las aplicaciones sigan funcionando aun cuando la carga de usuarios concurrentes es alta.  La ventaja de implementar en Heroku es que no es necesario tener en cuenta el sistema operativo subyacente, es decir, esta plataforma no requiere la instalación de actualizaciones, dependencias o el mantenimiento donde se ejecutarán las aplicaciones en producción. | |
| **Despliegue** | **Despliegues**  Los despliegues que utilizan infraestructura como servicio (IAAS) requieren la creación de servidores virtuales y la instalación de sistemas operativos como Linux o Windows con sus versiones y actualizaciones. Por esta razón, los analistas de datos deben preocuparse porque sus modelos y aplicaciones usen requisitos compatibles con el servidor de despliegue, reduciendo los tiempos de entrega de la solución.  Los ***dynos*** son contenedores aislados basados en sistemas operativos Linux que sirven como componentes básicos para el despliegue de modelos y aplicaciones. Se pueden crear de acuerdo con las necesidades del usuario, desde pequeños y económicos, los cuales ofrecen 1 procesador de CPU y 512 MB de capacidad de RAM, hasta contenedores más grande y costosos con procesadores de CPU ilimitados y hasta 32 GB de RAM. | |

|  |
| --- |
| Cuadro de texto |
| Para reforzar y profundizar en este tema, lo invitamos a leer el siguiente artículo.***” Automated Framework for Screening of Glaucoma Through Cloud Computing”*** Soorya, M., Issac, A., & Dutta, M. (2019). Recuerde que también se encuentra disponible en el material complementario:<https://www-proquest-com.bdigital.sena.edu.co/docview/2203597397?accountid=31491&pq-origsite=primo> |

## Despliegue de la API con contenedores

|  |  |
| --- | --- |
| **Tipo de recurso** | Tarjetas Animadas |
| **Introducción** |  |
| Imagen: 228131\_i1938 | Los contenedores se crearon, en principio, para implementar juegos en línea, debido a que reducen la sobrecarga del­ sistema, tienen tiempos de implementación más rápidos y permiten a los equipos de desarrollo pensar menos en la implementación y más en el contenido en producción. Por lo tanto, los creadores de juegos como Riot Games de League of Legends, los incorporan en sus proyectos. |
| Imagen: 228131\_i1939 | Los contenedores son la mejor opción para la virtualización. Estos virtualizan el sistema operativo, por lo general Linux, en lugar del hardware, dejando que cada contenedor individual almacene el programa, sus bibliotecas y dependencias. Debido a que no siempre necesitan un sistema operativo huésped, como lo hacen las máquinas virtuales, los contenedores pueden hacer uso de las capacidades y recursos del sistema operativo host, ya que son pequeños, rápidos y portátiles. Entre los más usados están Docker y Kubernetes. |
| Imagen: 228131\_i1940 | Los contenedores brindan a los desarrolladores la misma capacidad para aumentar el uso de CPU y memoria que las máquinas virtuales. Sin embargo, los contenedores van un paso más adelante, ya que, también admiten diseños de microservicios que permiten la utilización y escalado más preciso de los componentes de la aplicación. Debido a que solo un componente soporta la carga, este es un método rentable para ampliar las aplicaciones monolíticas. |
| Imagen: 228131\_i1941 | Usando los contenedores, los desarrolladores pueden crear e implementar aplicaciones de forma rápida y frecuente gracias a las mejoras en los métodos de integración y a las entregas continuas, cada vez más usuales en los proyectos de desarrollo basados en metodologías ágiles. |

## Docker

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Slider Presentación | |
| **Introducción** | Docker es una herramienta útil para empaquetar, enviar y ejecutar aplicaciones dentro de contenedores. Los contenedores y las máquinas virtuales tienen un objetivo afín: separar las aplicaciones de sus dependencias en unidades aisladas que puedan ejecutarse en cualquier entorno. Además, los contenedores y las máquinas virtuales ponen fin al requisito del hardware, lo que permite un uso más eficiente de recursos como memoria, CPU y disco, Así mismo minimizan el consumo de energía y los costos de mantenimiento. | |
| La principal diferencia entre Docker y máquinas virtuales está en su enfoque arquitectónico. Una máquina virtual es, básicamente, una emulación de un computador que ejecuta todo tipo de programas o aplicaciones. Las máquinas virtuales se ejecutan en la parte superior de una máquina física mediante un hipervisor. Por su parte, Docker no emula un sistema operativo, sino que está compuesto por el kernel y los componentes esenciales que soportan las aplicaciones para las que fue diseñado. | | Concepto centro de procesamiento de grandes datos, base de datos en la nube, futuro de estación de energía de servidor. Tecnología de transmisión de datos. Sincronizando información personal. Cadena de cubos o cajas de datos financieros abstractos  Imagen: 228131\_i1942 |
| Docker es una de las herramientas más utilizadas por los desarrolladores de software y analistas de datos para enviar a producción las aplicaciones y modelos. Incluso gigantes tecnológicos como Amazon, Google, IBM, Microsoft y VMware desarrollan servicios para soportar Docker en sus plataformas. | | St Petersburg, Russia - 23 Feb 2021: A man in a shirt sits in a modern office and uses a laptop to work in business or programming, application development  Imagen: 228131\_i1943 |
| La tecnología Docker se basa, principalmente, en la arquitectura cliente-servidor. El cliente se comunica con Docker, que actúa como un servidor presente en la máquina anfitriona. El servicio de Docker funciona con tres procesos principales que ejecutan, construyen y distribuyen contenedores. Tanto el contenedor Docker como el servicio pueden estar ubicados en una sola máquina. (Potdar et al., 2020) | | Concepto vector modelo de servidor cliente y almacenamiento en la nube seguro en Internet. Diversos datos de diferentes usuarios en almacenamiento en la nube con fácil acceso a cualquier tecnología de dispositivos en estilo plano  Imagen: 228131\_i1944 |
| Los contenedores Docker son creados de una imagen que contiene únicamente los requerimientos necesarios para ejecutar la aplicación deseada, por ejemplo, si se quiere desplegar un modelo de *machine learning* en una aplicación web, la imagen del contenedor Docker deberá tener instalado, como mínimo, el lenguaje de programación del modelo, ya sea Python, R o Java. Además, un servidor web como Apache o Nginx y las librerías requeridas para que la aplicación o modelo funcione correctamente. | | SaaS - Software como servicio. Línea de código de programación de aplicaciones de Internet. Software de nube en equipos con código de programa en pantalla, icono de elementos infográficos, aplicación, pantallas virtuales en blanco  Imagen: 228131\_i1945 |
| La arquitectura de Docker es muy práctica para realizar pruebas, ya que permite clonar las imágenes de una misma aplicación o modelo y ejecutarlas al tiempo en el mismo servidor, como lo ilustra la siguiente figura: | | **Figura 4** *Arquitectura Docker*  .  Imagen: 228131\_i1946 |
| En la figura se muestra que los contenedores empaquetan solo el espacio que el usuario requiere para el despliegue de la aplicación o modelo, y no el hardware, como lo haría una máquina virtual. Cada contenedor tiene su propio espacio aislado dentro del sistema para permitir que varios de ellos se ejecuten en una sola máquina host. Podemos ver que toda la arquitectura del sistema operativo se comparte entre contenedores, que junto a las bibliotecas, son las únicas partes que se crean desde cero, razón por la cual son tan livianos. | | Colocar imagen relacionada con el texto  **Colocar la imagen anterior** |

|  |  |
| --- | --- |
| **Tipo de recurso** | Rutas / Pasos. Verticales 1 |
| **Introducción** | Docker es una de las tecnologías a tener en cuenta en el proceso de despliegue de cualquier aplicación o modelo de inteligencia artificial. Las siguientes son algunas de sus ventajas: |
| Imagen general que ilustre el tema    Imagen: 228131\_i1947 | |
| **Botón 1** | **Facilidad de uso**:  Docker ha brindado a desarrolladores, analistas de datos, administradores de sistemas y arquitectos la creación fácil y eficiente de contenedores para probar rápidamente aplicaciones portátiles. Además, permite que cualquier usuario empaquete las aplicaciones en un computador portátil y de igual forma, pueda implementarlas sin mayores ajustes en cualquier nube privada, pública o servidor local. Lo ideal con Docker es construir una vez y ejecutar en cualquier lugar. |
| **Botón 2** | **Velocidad:**  Los contenedores Docker son livianos y rápidos, dado que se ejecutan en entornos separados del kernel del sistema operativo y, por tanto, hacen uso de pocos recursos. Así mismo, su creación y ejecución se puede realizar en pocos segundos en comparación con las máquinas virtuales que requieren más tiempo para iniciar el sistema operativo y cargar todos sus servicios y componentes. |
| **Botón 3** | **Docker Hub**:  Los usuarios de esta herramienta tienen a su disposición la variedad de posibilidades de Docker Hub, un almacén de aplicaciones para imágenes de Docker que cuenta con un gran número de imágenes públicas creadas y soportadas por la comunidad que la conforma, fáciles de buscar, que cumplen con los requerimientos y están listas para desplegar y usar sin modificaciones. |
| **Botón 4** | **Modularidad y flexibilidad**:  Docker permite la división de aplicaciones en módulos o contenedores, de modo que, se puede implementar la base de datos de una aplicación en un contenedor y la interfaz o visualización en otro. Con Docker, también se pueden integrar varios componentes o partes del sistema en un solo contenedor y con ello tener un único archivo para la base de datos, el *backend* y la interfaz de usuario, lo cual ayuda al mantenimiento futuro de la aplicación. |

|  |  |
| --- | --- |
| **Tipo de recurso** | Infografía estática |
| **Texto introductorio** | Docker es una herramienta muy utilizada para desplegar aplicaciones y modelos de analítica de datos. |
| **Imagen**  **Figura 5** *Funcionamiento Docker* | |
| **Código de la imagen** | 228131\_i1948 |

|  |
| --- |
| Cuadro de texto |
| Para reforzar y profundizar los conceptos en este tema, lo invitamos a leer el e-book.***” Primeros pasos y puesta en práctica de una arquitectura basada en micro-servicios”***) Gouigoux, J. (2018. Recuerde que también se encuentra disponible en el material complementario:<https://www-proquest-com.bdigital.sena.edu.co/docview/2203597397?accountid=31491&pq-origsite=primo> |

# Implementación en IAAS (AWS ECS)

|  |  |
| --- | --- |
| Tipo de recurso | Cajón de texto de color |
| IAAS es una plataforma donde un proveedor ofrece infraestructura tecnológica por un determinado costo de acuerdo al hardware contratado. En este enfoque el despliegue de máquinas virtuales es sencillo y solo requiere de unos cuantos clics para configurar las características como CPU, memoria RAM, disco duro y red. Centro de datos: Administrador de sistemas femenino y charla de especialista en TI masculino, use laptop. Los ingenieros de tecnología de la información trabajan en la protección de la ciberseguridad en la granja de servidores de computación en nube.  En estas plataformas existen imágenes precargadas que facilitan la elección del sistema operativo, bases de datos, firewall y demás herramientas necesarias para el despliegue de las aplicaciones y modelos de *machine learning*.  El proveedor suministra una máquina y un sistema operativo, pero los usuarios tienen que instalar todos los paquetes de software y configurar los tiempos de ejecución de servidores y bases de datos que requieran las aplicaciones. En general, IAAS requiere un equipo para administrar el sistema, aplicar reglas de firewall y parches de seguridad con frecuencia.  Imagen: 228131\_i1949 | |

## Introducción a AWS

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | | Pestañas o tabs Verticales |
| **Introducción** | | AWS o Amazon Web Service es una de las plataformas más populares y completas del mercado, la cual ofrece más de 200 servicios de computación y almacenamiento en la nube, fue lanzada el año 2006 y pertenece al holding de empresas de Amazon.  A nivel de infraestructura, esta plataforma está compuesta por regiones, zonas de disponibilidad y data centers. A continuación, se describen estos conceptos: |
| Rostov-on-Don, RUSSIA - February 15 2021: Amazon Web Services logo on the smartphone screeni  Imagen: 228131\_i1950 | | |
| **Regiones** | **Regiones**:  Una región es una zona geográfica física en donde AWS ha implementado sus *datacenters*. Cabe destacar que una región NO es un país y que cada una de ellas está conformada por zonas de disponibilidad. | |
| **Zonas de disponibilidad** | **Zonas de disponibilidad**:  Conformadas por 2 o más datacenters ubicados en la misma zona pero aislados geográficamente de manera estratégica. | |
| **Datacenters** | **Datacenters**:  Cada uno de los *datacenter* tiene su propia fuente de alimentación, ventilación y conectividad de red, estos se encuentran separados por aproximadamente 100 km pero al mismo tiempo están interconectados entre sí para apoyar la zona de disponibilidad en una red redundante que ofrece latencias muy bajas. | |
| **Servicios AWS** | **Servicios AWS**  Para acceder a los servicios de AWS es necesario crear una cuenta a través del diligenciamiento de un formulario con los datos personales, el email del usuario y una tarjeta de crédito para la verificación de identidad y cobros futuros.  AWS brinda una capa gratuita que permite el acceso a 18 productos y servicios durante los primeros 12 meses, sin embargo, esta capa tiene límites de uso que restringen características esenciales de las máquinas virtuales como la capacidad de disco duro, memoria RAM y CPU.  AWS incorpora la inteligencia artificial y los algoritmos de *machine learning*, campos relacionados con la analítica de datos que ofrecen múltiples posibilidades y beneficios para que las empresas puedan obtener predicciones acertadas (Serrano, 2022). | |

|  |  |
| --- | --- |
| **Tipo de recurso** | Infografía estática |
| **Texto introductorio** | AWS proporciona una serie de ventajas para las organizaciones que deseen migrar sus servicios a la nube. |
| **Imagen**  **Figura 6** *Infografía servicios AWS* | |
| **Código de la imagen** | 228131\_i1951 |

## Costos y precauciones de AWS

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Carrusel de tarjetas | |
| **Introducción** | La filosofía de costos de AWS es pagar solo por lo que se utiliza, lo que supone una ventaja en comparación con otros tipos de plataformas en las que se realizan cobros por los productos y servicios que se ponen a disposición del usuario aun sin hacer uso de ellos. | |
| Imagen general que ilustre el tema (opcional)  SaaS - Software como servicio, a petición. Concepto de Internet y tecnología en pantalla virtual.i  Imagen: 228131\_i1952 | | |
| AWS cuenta con una característica de administración de costos que se puede utilizar para visualizar y analizar los datos detalladamente, de esta manera es posible estar al día con las tendencias y predecir de manera efectiva los costos futuros estimados de Amazon Web Services. | | Análisis de datos de concepto de diseño plano. Visualice con un gráfico y un gráfico de crecimiento de marketing. Ilustración vectorial.  Imagen: 228131\_i1953 |
| AWS permite almacenar terabytes o petabytes de datos sin generar un alto costo, sin embargo, cuando se hacen transferencias desde estos almacenamientos hacia otras instancias el costo se eleva considerablemente, por lo es importante tener cuidado con las transferencias y descargas de los datos. | | i  Imagen: 228131\_i1954 |
| Beneficios como la alta disponibilidad y escalabilidad son atractivos para cualquier organización que desea brindar a sus usuarios servicios de calidad y sin interrupciones, sin embargo, contratar servicios de redundancia con AWS genera costos adicionales en la facturación que deben analizarse previamente para equilibrar la balanza costo/beneficio | | Imagen: 228131\_i1955 |
| Para minimizar los costos en AWS se recomienda identificar y terminar las instancias y recursos no utilizados o mover los datos poco o no utilizados hacia los niveles AWS destinados para almacenamiento en frio. También es preferible utilizar sistemas operativos de código abierto como Linux, ya que otros, como los de la familia Microsoft, generan costos de licenciamiento elevados. | | i  Imagen: 228131\_i1956 |

## Manejo de AWS

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Pestañas o tabs horizontales | |
| **Introducción** | El acceso y manipulación de la plataforma de Amazon Web Services se puede realizar a través de su consola de administración, la CLI y AWS SDKs o Kit de desarrollo de software. Cada método tiene sus utilidades, ventajas, desventajas y tipo de usuario. A continuación, se detalla cada uno de ellos: | |
| **Consola de administración AWS** | **Consola de administración AWS**  Es una interfaz gráfica fácil de utilizar mediante el navegador. La consola de administración es una aplicación web que agrupa y hace mención a un enorme conjunto de servicios para el manejo de recursos de AWS.  La consola de administración de AWS también se encuentra disponible para diferentes sistemas operativos y plataformas móviles como Android e iOS, lo cual permite que los encargados de su administración puedan ver y manejar de forma fácil los recursos y gestionar las alarmas que requieran atención desde su teléfono móvil. | Imagen: 228131\_i1957 |
| **AWS CLI** | **AWS CLI**  Proporciona acceso a la plataforma a través de una consola de comandos discretos. Este acceso permite controlar todos los servicios de Amazon Web Services por medio de líneas de comandos.  La interfaz de línea de comandos (AWS CLI) es una herramienta *open source* que permite manipular los servicios de AWS mediante el uso de comandos en una aplicación que los interpreta y envía las órdenes a la plataforma. Todas las funciones de administración, gestión y acceso de IaaS (infraestructura como servicio) de la AWS Management Console están disponibles en la AWS API y la AWS CLI. | Imagen: 228131\_i1958 |
| **AWS SDK:** | **AWS SD**  Esté acceso a la plataforma de AWS incorpora conectividad, funcionalidades de la nube de AWS y facilita el acceso mediante los diversos lenguajes de programación. AWS SDK (kit de desarrollo de software) ayuda a la codificación de diferentes lenguajes de programación para la interacción con los servicios de AWS. Existen diferentes kits de desarrollo dependiendo del lenguaje de programación, entre los más populares están Javascript, Java y Python | Imagen: 228131\_i1959 |

## Despliegue en ECS

|  |  |
| --- | --- |
| **Tipo de recurso** | Tarjetas Animadas |
| **Introducción** | ECS o Elastic Container Service es un servicio de la plataforma AWS que permite la orquestación de contenedores completamente administrados para la implementación y escalabilidad de aplicaciones. ECS ofrece las siguientes características para potenciar los despliegues de aplicaciones y modelos de *machine learning* en la nube: |
| Imagen: 228131\_i1960 | **Compatibilidad**  Amazon ECS es compatible con Docker y permite la ejecución e integración con Docker Compose y sus repositorios desde Docker Hub. Además, se integra con contenedores de Windows de forma optimizada |
| Imagen: 228131\_i1961 | **Administración**  Amazon ECS permite, a través de un conjunto de APIs, la integración y ampliación de los servicios desplegados en la plataforma. Las APIs facilitan la creación y eliminación de *clústeres* y el lanzamiento y terminación de contenedores Docker. |
| Imagen: 228131\_i1962 | **Programación de tareas**  Amazon ECS incorporaestrategias de programaciónpara administrar los contenedores de los diferentes *clústeres* con base en las necesidades de los recursos de hardware como CPU o RAM y los requerimientos de disponibilidad. |
| Red informática  Imagen: 228131\_i1963 | **Redes**  Amazon ECS permite equilibrar el tráfico y conectividad de los contenedores con la implementación de balanceadores de carga de redes. Esto es muy útil para garantizar la disponibilidad de aplicaciones consumidas por muchos usuarios de forma concurrente. |
| Imagen: 228131\_i1964 | **Monitoreo**  Amazon ECS, a través de la herramienta CloudWatch, proporciona a los usuarios la posibilidad de visualizar el registro de las ejecuciones y uso de los recursos de los contenedores para realizar los respectivos ajustes de seguridad a las aplicaciones y soluciones desplegadas. |

|  |
| --- |
| Cuadro de texto |
| Para reforzar y profundizar los conceptos de este tema, lo invitamos a ver el siguiente video***: “Webinar, Managing Big Data Business Intelligence on the AWS Cloud”.*** Amazon Web Services, & YouTube (2019). Recuerde que también se encuentra disponible en el material complementario:<https://www-virtualpro-co.bdigital.sena.edu.co/biblioteca/seminario-web-gestionando-inteligencia-de-negocios-de-datos-masivos-sobre-aws-cloud> |

**Síntesis**

|  |  |
| --- | --- |
| **Tipo de recurso** | Síntesis |
| Nombre del **mapa**: Despliegue de solución informática  Síntesis: | |
| **Introducción** | En el **siguiente** mapa podrá apreciar un resumen de los temas abordados con los conceptos básicos de despliegue de solución informática: |
| **Figura 7** *Síntesis despliegue de la solución informática*    Imagen: 228131\_i1965  Ruta de Acceso al archivo original: <https://mm.tt/map/2325418452?t=BBIbUUFf1r> | |

|  |
| --- |
| Cuadro de texto |
| Estimado aprendiz, ha llegado al final de este componente formativo. Es hora de resolver la actividad de aprendizaje y revisar el material complementario para afianzar sus conocimientos. |

**Actividad didáctica**

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Actividad didáctica. Verdadero y falso | |
| Apreciado aprendiz, a continuación, encontrará una actividad didáctica, para afianzar los conceptos y aprendizajes del componente de formación sobre despliegue de solución informática.  Antes de realizarlo, se recomienda la lectura del componente formativo mencionado. Esta actividad no es calificable, y puede realizarse todas las veces que se desee.  Lea atentamente las preguntas y seleccione, para cada una de ellas, si considera que el enunciado es verdadero o falso. | | Imagen: 228131\_i1966 |
| Un pipeline de datos es un conjunto de etapas y herramientas tecnológicas que se agrupan para ejecutar los procesos de transformación de los datos desde el inicio hasta su almacenamiento persistente. | | El sistema de administración de documentos (DMS) está siendo configurado por un consultor de TI que trabaja en un equipo portátil en la oficina. Software para archivar, buscar y administrar archivos e información corporativos. Procesos empresariales  Imagen: 228131\_i1967 |
| Verdadero (correcto) | | Falso |
| **Retroalimentación opción correcta:** ¡Excelente! Los Pipelines permiten optimizar las tareas desde el inicio hasta su terminación. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| Las consultas de datos en un pipeline se deben realizar al final de la tubería y por lo tanto son consultas estáticas. | | Código SQL (Lenguaje de consulta estructurado) en el monitor de equipo con fondo de la base de datos y de la sala de servidores. Ejemplo de código SQL para consultar datos de una base de datos.  Imagen: 228131\_i1968 |
| Verdadero | | Falso (correcto) |
| **Retroalimentación opción correcta:** ¡Excelente! Los Pipelines permiten que los analistas de datos consulten las tablas y esquemas interactivamente sin necesidad de esperar el final de la tubería para mostrarlas. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| Los lagos de datos proporcionan a los analistas de la capacidad de procesar información desde distintos volúmenes de datos con altos niveles de tolerancia a fallas. | | Panel de resultados de datos de KPI financieros en el equipo  Imagen: 228131\_i1969 |
| Verdadero (correcto) | | Falso |
| **Retroalimentación opción correcta:** ¡Excelente! Los lagos de datos se alimentan desde distintas fuentes con datos estructurados y no estructurados. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| Se conoce como ingeniería de requerimientos a una serie de transformaciones que se realizan previo al ingreso de los datos al modelo de *machine learning*. | | requisitos del proyecto. analista de negocios que crea requerimientos de proyectos, empresa de TI, proceso de desarrollo de estrategias, concepto de ingeniero de implementación.  Imagen: 228131\_i1970 |
| Verdadero | | Falso (correcto) |
| **Retroalimentación opción correcta:** ¡Excelente! La ingeniería de requerimientos no realiza transformaciones a los datos, esto es propio de la ingeniería de funciones. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| La imputación de datos faltantes es un proceso donde se reemplazan valores inexistentes por números obtenidos de estimaciones estadísticas. | | documentos digitales y archivos de datos compartidos con almacenamiento en la nube  Imagen: 228131\_i1971 |
| Verdadero (correcto) | | Falso |
| **Retroalimentación opción correcta:** ¡Excelente! En la imputación de datos se reemplazan los faltantes por la media o la moda estadística del conjunto de los datos. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| El estándar PMML es ideal para la exportación de modelos de *Deep Learning* | | Diseño basado en el concepto de aprendizaje profundo con ilustración isométrica del brillante cerebro humano con ruedas cognoscitivas sobre el fondo de la matriz o la codificación binaria.  Imagen: 228131\_i1972 |
| Verdadero | | Falso (Correcto) |
| **Retroalimentación opción correcta:** ¡Excelente! El estándar PMML o *Predictive Model Markup Language* no es ideal para la exportación de modelos de *Deep Learning* sino para soportar modelos estadísticos y de minería de datos. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| **REST** Es una interfaz que permite la conexión entre varios sistemas que utilizan el protocolo HTTP. | | Interfaz de programación de aplicaciones API conectar servicios en Internet y permitir la comunicación de datos de red, el concepto de toques de ingeniero de software para IoT, cloud computing, automatización de procesos robótica  Imagen: 228131\_i1973 |
| Verdadero (Correcto) | | Falso |
| **Retroalimentación opción correcta:** ¡Excelente! REST o la Transferencia de Estado de Representación facilita a los usuarios la interacción con la aplicación. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| **Heroku** es una infraestructura como servicio (IaaS) que se ejecuta localmente y cuenta con capacidad para soportar algunos lenguajes de programación. | | IaaS, Infrastructure as a Service. Internet and networking concept.  Imagen: 228131\_i1974 |
| Verdadero | | Falso (Correcto) |
| **Retroalimentación opción correcta:** ¡Excelente! Heroku no es una Infraestructura como servicio (IAAS), sino una plataforma como servicio (PAAS) | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| La tecnología Docker se basa principalmente en la arquitectura cliente-servidor. | | Vector de personas de negocios que trabajan en computadoras usando el servicio en la nube  Imagen: 228131\_i1975 |
| Verdadero (Correcto) | | Falso |
| **Retroalimentación opción correcta:** ¡Excelente! Los clientes realizan peticiones al Docker que opera como servidor. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |
| AWS CLI Es una Interfaz gráfica fácil de utilizar mediante el navegador. | | LISBON, PORTUGAL - FEBRUARY 19, 2014: Amazon Web Services homepage through a magnifying glass. AWS is a collection of web services that together make up a cloud computing platform.  Imagen: 228131\_i1976 |
| Verdadero | | Falso (Correcto) |
| **Retroalimentación opción correcta:** ¡Excelente! AWS CLI no es una interfaz gráfica, sino una Interfaz de línea de comandos. | | **Retroalimentación opción incorrecta:** Opción incorrecta, se recomienda volver a revisar el componente formativo e intentar nuevamente. |

**Material complementario**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de recurso** | Material complementario | | |
| **Tema** | **Referencia APA del material** | **tipo** | **Enlace** |
| AWS | Amazon Web Services, & YouTube. (2019), *Webinar, Managing Big Data Business* Intelligence on the AWS Cloud. | Video | <https://www-virtualpro-co.bdigital.sena.edu.co/biblioteca/seminario-web-gestionando-inteligencia-de-negocios-de-datos-masivos-sobre-aws-cloud> |
| Docker | Gouigoux, J. (2018), Docker : *Primeros pasos y puesta en práctica de una arquitectura basada en micro-servicios (Epsilon)*. Barcelona: Ediciones ENI. | Libro | <https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1j5choe/sena_biblioteca_eniEPT2DOC> |
| Heroku | Soorya, M., Issac, A., & Dutta, M. (2019). *Automated Framework for Screening of Glaucoma Through Cloud Computing*. Journal of Medical Systems, 43(5), 136-17. | Artículo | <https://www-proquest-com.bdigital.sena.edu.co/docview/2203597397?accountid=31491&pq-origsite=primo> |
| Implementación de Pipeline | Tuychiev, Bex. (2020), *How to Use Sklearn Pipelines For Ridiculously Neat Code*. Towards Data Science. | Artículo | <https://towardsdatascience.com/how-to-use-sklearn-pipelines-for-ridiculously-neat-code-a61ab66ca90d> |
| Optimización de la implementación de modelos | Ortiz Loaiza, A. (2020), *Optimización del tiempo de inferencia de un modelo de machine learning usando OpenVINO y despliegue del modelo en un entorno cloud* | Artículo | <https://eprints.ucm.es/id/eprint/62126/1/ORTIZ_LOAIZA_Inference_time_optimization_on_a_machine_learning_model_using_OpenVINO_and_deployment_of_the_model_in_a_cloud_1603767.pdf> |

**Glosario**

|  |  |
| --- | --- |
| **Tipo de recurso** | Glosario |
| API | Acrónimo de Interfaz de Programación de Aplicaciones. Es un intermediario de software que permite que dos aplicaciones se comuniquen entre sí. |
| Datacenter | Centro de datos con recursos de computación de alta disponibilidad. |
| Dynos | Contenedores aislados basados en sistemas operativos Linux que sirven como componentes básicos para el despliegue de modelos y aplicaciones. |
| Ingeniería de funciones | Serie de transformaciones que se realizan previo al ingreso de datos al modelo de *Machine Learning*. |
| ONNX | El estándar ONNX (Open Neural Network Exchange) es un formato para crear diagramas de la red en un archivo binario. |
| Pipeline | Conjunto de fases y herramientas tecnológicas que se integran para realizar los procesos de transformación de los datos desde el inicio hasta el su almacenamiento persistente |
| PMML | El estándar PMML o *Predictive Model Markup Language* es un lenguaje de marcado que tiene como base a XML. |
| Región de AWS | Zona geográfica donde AWS ha creado varios Datacenters. |
| REST | Acrónimo de Transferencia de Estado de Representación. Es una interfaz que permite la conexión entre varios sistemas que utilizan el protocolo HTTP. |
| Zona de disponibilidad | Agrupación de 2 o más Datacenters asociados a una determinada región. |

**Referencias bibliográficas::**

|  |  |
| --- | --- |
| **Tipo de recurso** | Bibliografía |
| Cedeno-Mieles, V., Hu, Z., Ren, Y., Deng, X., Contractor, N., Ekanayake, S., . . . Self, N. (2020), *Data analysis and modeling pipelines for controlled networked social science experiments.* PloS One, 15(11), E0242453. Web <https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN_cdi_plos_journals_2464158716> | |
| Danielsson, P., Postema, T., & Munir, H. (2021), *Heroku-Based Innovative Platform for Web-Based Deployment in Product Development at Axis. IEEE Access, 9, 10805-10819*. Web <https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN_cdi_proquest_journals_2479888834> | |
| López García, &. (2019), *Deepaas Api: A Rest api for Machine Learning and Deep Learning models*. Journal of Open Source Software, 4(42), 1517. Web <https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN_cdi_crossref_primary_10_21105_joss_01517> | |
| Potdar, A., D G, N., Kengond, S., & Mulla, M. (2020), *Performance Evaluation of Docker Container and Virtual Machine. Procedia Computer Science,* 171, 1419-1428. Web <https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN_cdi_crossref_primary_10_1016_j_procs_2020_04_152> | |
| Serrano, Javier. "*AWS, El Rentable Negocio En La Nube De Amazon*." Actualidad Económica (Madrid, Spain) (2022): 17. Web. Web <https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN_cdi_proquest_miscellaneous_2664218413> | |