

# Resultados de Examen P2 para MAYANCELA CORDOVA JUAN FRANCISCO

! Las respuestas correctas están ocultas.

Puntaje para este examen: **82.17** de 100

Entregado el 1 de sep de 2023 en 14:52

Este intento tuvo una duración de 56 minutos.

## Pregunta 1

1 / 1 pts

¿Cómo pueden funcionar las redes definidas por software (SDN) con la computación de borde (Edge)?

- ☐ La computación perimetral no funciona en una red definida por software
- ☒ SDN puede optimizar la forma en que la computación de borde procesa los datos
- ☐ Los datos de una SDN solo funcionan en la red, y no son enviados a una SDN
- ☐ Edge Computing solo puede procesar datos de una red tradicional y no de una SDN.

## Pregunta 2

1 / 1 pts

Componentes de OpenStack. Unir según corresponda:

<b>Ceilometer</b>	Proporciona los servicios de t 
<b>Heat</b>	Proporciona un servicio o fun 
<b>Glance</b>	Proporciona servicios de imá 
<b>Horizon</b>	Proporciona una GUI de adm 

Parcial

**Pregunta 3**

0.67 / 1 pts

¿Cuáles de las siguientes son las limitaciones de IaaS? Escoja los que considere.

- ☒ Débil seguridad para múltiples inquilinos
- ☒ Sistemas heredados que operan en la nube
- ☐ Seguridad débil
- ☐ Automatiza el despliegue de almacenamiento y servidores.

**Pregunta 4****1 / 1 pts**

¿Cuáles de las siguientes son las características de IaaS? Escoja los que considere.

- ☐ Débil seguridad para múltiples inquilinos
- ☒ Es altamente escalable
- ☒ Facilita el acceso de un solo hardware a muchos usuarios
- ☒ Es una plataforma flexible y dinámica

**Parcial****Pregunta 5****0.5 / 1 pts**

En el procesamiento de datos IoT con enfoque en la nube existen un sinnúmero de tareas que se desempeñan en capas distintas. **Agrupar sus opciones según corresponda**

Cloud Computing	<div>big data</div> <div>data warehouse</div>
Fog Computing	<div>gateway</div>

	redes locales
Edge Computing	sistemas embebidc virtualizacion

**Opciones disponibles:**

- Big Data
- Data warehouse
- Virtualización
- Redes Locales
- Gateway
- Sistemas embebidos

**Respuesta 1:**

Big Data

**Respuesta 2:**

Data warehouse

**Respuesta 3:**

gateway

**Respuesta 4:**

## Redes Locales

### Respuesta 5:

sistemas embebidos

### Respuesta 6:

virtualizacion

## Pregunta 6

1 / 1 pts

Procesador es un componente principal en NiFi, que realmente funcionará en el contenido de FlowFile y ayuda a crear, enviar, recibir, transformar el enrutamiento, dividir, fusionar y procesar FlowFile.

### Respuesta 1:

Procesador

## Pregunta 7

1 / 1 pts

En una arquitectura de IoT para el procesamiento de datos, ¿Cuales de las siguientes funciones se deben realizar en la capa de borde (Edge Layer)?

☒ Puertas de enlace

☐ Procesamiento de Big Data☒ Sistemas embebidos☐ Análisis de datos & Reducción

## Pregunta 8

1 / 1 pts

¿Cuáles son los beneficios de la seguridad en edge computing?

☐ La seguridad de la computación de borde solo beneficia a IoT☒ La seguridad puede responder en tiempo real y alojar análisis de amenazas de comportamiento☒ Edge computing solo protege los datos sobre los bordes.☐ La computación de borde por definición no puede ser segura

## Pregunta 9

1 / 1 pts

¿Cuales de las siguientes opciones indican la importancia del análisis de datos en IoT?

- ☒ Salvaguarda la seguridad y protección de personas y propiedades
- ☒ Conduce a la producción de aplicaciones y servicios pioneros y centrados en las personas.
- ☒ Ayuda a obtener predicciones y prescripciones precisas,
- ☐ Permite establecer la acción en solo un campo/entorno del conocimiento.

### Pregunta 10

1 / 1 pts

Las imágenes de VM disponibles a través de Glance se pueden almacenar en una variedad de ubicaciones, desde simples sistemas de archivos hasta sistemas de almacenamiento de objetos como el proyecto OpenStack Swift.

#### Respuesta 1:

Glance

### Pregunta 11

10 / 10 pts

Explique **DETALLADAMENTE** cómo funciona *BLOCKCHAIN* y el mecanismo de *HASHING* que utiliza para mantener inmutabilidad en toda la cadena de bloques.

Su respuesta:

**Funcionamiento de Blockchain:**

Blockchain es una cadena de bloques compuesta por una serie de bloques de datos. Cada bloque contiene una cantidad de registros de transacciones y una marca de tiempo que indica cuándo se agregó el bloque a la cadena. Cada bloque contiene un enlace al bloque anterior mediante un valor conocido como "hash previo". Esto crea una secuencia continua de bloques, donde cada bloque está conectado al anterior, formando una cadena. La cadena de bloques se almacena en una red distribuida de computadoras llamadas nodos. Cada nodo tiene una copia completa de la cadena de bloques y verifica y valida las transacciones antes de agregarlas a un bloque. Cada bloque incluye un valor de hash generado a partir de los datos del bloque y el hash del bloque anterior. Este valor de hash actúa como una huella digital única para ese bloque y su contenido. Si alguien intenta alterar el contenido de un bloque, el valor de hash cambiará y se detectará inmediatamente como una anomalía en la cadena. Esto garantiza la integridad de la cadena de bloques.

**Mecanismo de Hashing:**

Cada conjunto de datos produce un hash único. Incluso una pequeña modificación en los datos de entrada generará un hash completamente diferente. La misma entrada siempre generará el mismo hash. Esto asegura la coherencia en la generación de hashes. La función de hash debe ser eficiente en términos de tiempo de procesamiento para que la generación de hashes sea rápida. No se puede obtener la entrada original a partir del hash. Es decir, el proceso de hashing es unidireccional, lo que garantiza la privacidad y la seguridad de los datos. La función de hash debe ser capaz de manejar cualquier tipo de dato de entrada, ya sean texto o números.

**Pregunta 12****18 / 20 pts**



El Departamento de Soporte Técnico (DST) ha recibido un requerimiento por parte del departamento de talento humano (DTH) para indexar y realizar búsquedas en todos los documentos personales en formato *pdf* de cada uno de los empleados. El ingeniero del DST se ha acordado que una herramienta vista en su curso de SISTEMA EN LA NUBE que le ayudaría a realizar aquello... Sin embargo, él sólo recuerda el nombre de la tecnología **ELASTICSEARCH**. ¿Le ayudaría a recordar?

Se pide, **INDICAR Y DESCRIBIR DETALLADAMENTE LOS 5 COMPONENTES QUE EL INGENIERO DEBE RECORDAR ACERCA DE ELASTICSEARCH PARA REALIZAR EL INDEXAMIENTO DE DATOS/ARCHIVOS. (4 ptos. c/u)**

Su respuesta:

**Índices:** En Elasticsearch, los datos se almacenan en índices. Un índice es una colección lógica de documentos que comparten una estructura y un esquema común.

**Documentos:** Los documentos son las unidades de datos básicas en Elasticsearch. En este contexto, un documento sería un archivo PDF de un empleado que se desea indexar y buscar. Los documentos se almacenan en formato JSON y contienen campos y valores que describen el contenido del archivo.

**Mapping:** El mapeo define la estructura de los documentos en un índice, especificando qué campos existen y qué tipo de datos contienen. Para indexar archivos PDF de empleados, el ingeniero del DST debe definir un mapeo que identifique los campos relevantes en los documentos PDF, como nombre, fecha de ingreso o número de empleado.

**Análisis de texto:** Elasticsearch proporciona capacidades de análisis de texto para indexar y buscar texto en documentos. Esto es esencial cuando se trata de archivos PDF, ya que necesita extraer y analizar el texto contenido en los archivos PDF para que sea searchable. Elasticsearch ofrece analizadores que pueden dividir el texto en términos (tokens).

**Consultas y búsquedas:** Una vez que los documentos están indexados en Elasticsearch, se pueden realizar consultas y búsquedas para recuperar información relevante. Elasticsearch ofrece una variedad de consultas, como consultas de coincidencia, consultas de rango, que permiten buscar y filtrar documentos de manera eficaz.

### Pregunta 13

15 / 15 pts

Describa **DETALLADAMENTE** los componentes **básicos** que son necesarios para una implementación/instalación de **OPENSTACK**. (2pto c/u)

Su respuesta:

**Keystone:** Administra la autenticación de usuarios y servicios, así como la asignación de roles y permisos. Keystone también emite tokens de autenticación que se utilizan para acceder a los servicios de OpenStack.

**Glance:** servicio encargado de gestionar y almacenar imágenes de máquinas virtuales, que se utilizan como plantillas para crear instancias de máquinas virtuales.

**Nova:** Permite la creación, el inicio, la detención y la eliminación de instancias de VM. Nova también gestiona la asignación de recursos computacionales, como CPU y memoria.

**Neutron:** Permite la creación de redes virtuales, subredes y enrutadores para conectar las instancias de máquinas virtuales. También facilita la configuración de firewalls y reglas de seguridad.

**Cinder:** Permite crear y adjuntar volúmenes de almacenamiento a las instancias. Los volúmenes pueden ser escalados y gestionados de forma independiente.

**Swift:** Permite el almacenamiento y la recuperación de datos no estructurados, como imágenes, vídeos y documentos, a través de una API RESTful.

**Horizon:** Permite gestionar recursos de forma gráfica, como la creación de instancias, redes y almacenamiento.

**MariaDB o MySQL:** OpenStack utiliza una base de datos relacional para almacenar información de configuración y estado. MariaDB o MySQL son sistemas de gestión de bases de datos comúnmente utilizados para este propósito.

**RabbitMQ:** OpenStack utiliza un servicio de cola para gestionar la comunicación entre sus componentes. RabbitMQ es una opción común para garantizar la entrega de mensajes entre los servicios de OpenStack.

## Pregunta 14

20 / 20 pts

Explique **DETALLADAMENTE** el funcionamiento de **APACHE KAFKA** y para que se utilizan los cuatro componentes básicos en un canal de comunicación ó tópico de Apache Zookeeper. **(5 ptos. c/u)**

Su respuesta:

**Productor:** Los productores son responsables de enviar datos a los tópicos de Kafka. Los productores producen registros o mensajes y los publican en un tópico específico.

**Tópico:** Un tópico es una categoría o canal de comunicación que representa una secuencia de registros o mensajes. Los tópicos son utilizados para organizar y etiquetar los datos en Kafka.

**Broker :** Los brokers de Kafka son servidores que gestionan el almacenamiento y la distribución de los datos. Los tópicos se dividen en particiones, y cada partición es replicada en múltiples brokers para garantizar la tolerancia a fallos y la escalabilidad.

**Consumidor** : Los consumidores son aplicaciones o sistemas que leen datos de los tópicos de Kafka y procesan los registros. Los consumidores pueden suscribirse a uno o varios tópicos y leer los mensajes en tiempo real o desde un punto de partida específico.

**El funcionamiento de Apache Kafka:**

- Los productores envían registros o mensajes a tópicos específicos.
- Los tópicos están divididos en particiones, y cada partición puede estar replicada en varios brokers.
- Los consumidores se suscriben a uno o varios tópicos y leen los mensajes de las particiones correspondientes.
- Kafka garantiza la tolerancia a fallos y la escalabilidad al distribuir las particiones en múltiples brokers.
- Los offsets de lectura se utilizan para realizar un seguimiento del progreso de lectura de los consumidores en cada partición.
- Kafka retiene mensajes durante un período configurable, lo que permite el procesamiento en lotes o la relectura de datos históricos.

**Pregunta 15**

10 / 10 pts

Estás diseñando una aplicación que necesita modelar y consultar relaciones complejas entre usuarios y sus interacciones en una red social. **Describe cómo podrías diseñar la base de datos utilizando Neo4j, incluyendo la estructura de nodos y relaciones, así como ejemplos de consultas Cypher que permitan recuperar información relevante de la red social. (2 Query) - Puede utilizar draw.io para que muestre mejor su diseño.**

Su respuesta:

Podemos diseñar un grafo que represente a los usuarios como nodos y las interacciones como relaciones entre esos nodos.

## Modelo de Datos:

- Cada usuario se representa como un nodo en el grafo. Cada nodo de usuario tiene propiedades como nombre de usuario, nombre completo, fecha de registro, ubicación, etc.
- Cada publicación realizada por un usuario se representa como un nodo de publicación. Cada nodo de publicación tiene propiedades como contenido, fecha de publicación, número de likes, etc.
- Cada comentario en una publicación se representa como un nodo de comentario. Cada nodo de comentario tiene propiedades como "contenido," "fecha de comentario," "usuario que lo hizo," etc.
- Las amistades entre usuarios se representan como relaciones. Por ejemplo, una relación AMIGO\_DE conecta dos nodos de usuario que son amigos. Esta relación puede tener propiedades como fecha de amistad.
- Cuando un usuario sigue a otro, se representa como una relación SIGUE\_A. Esto permite hacer un seguimiento de las relaciones de seguidores y seguidos.
- Cada nodo de usuario tiene relaciones salientes que apuntan a sus publicaciones. Estas relaciones se pueden llamar HA\_PUBLICADO y pueden tener propiedades como fecha de publicación.
- Cada nodo de usuario tiene relaciones salientes que apuntan a los comentarios que ha hecho en publicaciones. Estas relaciones se pueden llamar HA\_COMENTADO y pueden tener propiedades como "fecha de comentario."
- Cuando un usuario da "Me gusta" a una publicación, se representa como una relación LE\_GUSTA\_A. Esto permite hacer un seguimiento de las interacciones de Me gusta.

## ejemplos de consultas Cypher

### consulta 1

```
MATCH (usuario:Usuario {nombreUsuario: 'nombreUsuarioEjemplo'})-[:AMIGO_DE]-(amigo) OPTIONAL  
MATCH (amigo)-[:HA_PUBLICADO]-(publicacion) RETURN usuario, amigo, COLLECT(publicacion) AS  
publicacionesAmigo
```

Esta consulta encuentra un usuario por su nombre de usuario, identifica a sus amigos y recupera las publicaciones de esos amigos.

Consulta 2

MATCH (publicacion:Publicacion {idPublicacion: 'idPublicacionEjemplo'})-[:TIENE\_COMENTARIO]-(comentario) MATCH (usuario:Usuario)-[:HA\_COMENTADO]->(comentario) RETURN comentario, usuario  
Esta consulta busca una publicación por su identificador único, encuentra los comentarios en esa publicación y luego identifica a los usuarios que hicieron esos comentarios.

## Pregunta 16

0 / 15 pts

Imagina que estás diseñando una solución de Internet de las Cosas (IoT) que utiliza **EdgeX Foundry para gestionar dispositivos y datos en el borde**. Describe cómo podrías implementar un flujo de trabajo utilizando los servicios de EdgeX para recopilar datos de sensores en el borde, procesarlos y enviarlos a la nube para un análisis posterior. **Menciona los componentes clave de EdgeX que utilizarías en este flujo de trabajo.**

Su respuesta:

primenrammnete podria

1. Configuración de dispositivos: Utilizar el servicio Device Service para configurar y gestionar los dispositivos IoT en el borde. Registrar los dispositivos y definir sus perfiles y características.
2. Lectura de datos: Utilizar el servicio Core Data para recopilar lecturas de sensores desde los dispositivos registrados. Core Data almacena y gestiona los datos recopilados en una base de datos.
3. Reglas y transformaciones: Utilizar el servicio Core Metadata para definir reglas y transformaciones de datos. Esto permite normalizar, filtrar o transformar los datos recopilados según sea necesario.
4. Procesamiento en el borde: Utilizar el servicio Core Command para ejecutar acciones en el borde basadas en eventos específicos. Por ejemplo, ejecutar análisis preliminares o tomar decisiones en función de los datos recibidos.
5. Enrutamiento: Utilizar el servicio Export para enrutar los datos procesados hacia diferentes destinos, como la nube. Configurar un exportador que envíe los datos a una plataforma en la nube para su análisis y almacenamiento.
6. Integración con la nube: Utilizar el servicio App Functions SDK para desarrollar funciones personalizadas que se ejecuten en respuesta a eventos. Esto puede incluir la lógica de envío de datos a la nube o la realización de cálculos adicionales antes del envío.
7. Enviar a la nube: Utilizar el exportador configurado en el servicio Export para enviar los datos procesados a una plataforma en la nube para su almacenamiento y análisis adicional.