

Demo

Submit this form with test data, and Form Publisher will generate a test invoice for you that you will receive via email. You can then edit all the settings you want to customize your workflow.

¿Cuál es el propósito principal de JHipster?

1 punto

- ☒ Generar aplicaciones web Spring Boot con Angular, React o Vue de forma ágil
- ☐ Crear microservicios utilizando frameworks como Flask o Django
- ☐ Proporcionar una herramienta de integración continua para proyectos Spring Boot
- ☐ Generar aplicaciones Node.js con soporte para GraphQL

En JHipster, ¿qué es JDL (JHipster Domain Language)?

1 punto

- ☒ Un lenguaje específico de dominio para describir entidades y relaciones en una aplicación
- ☐ Un conjunto de herramientas para integración continua y despliegue
- ☐ Una interfaz de usuario para crear aplicaciones web de forma visual
- ☐ Una biblioteca para la generación automática de pruebas unitarias

¿Qué ventaja proporciona JHipster a la hora de crear aplicaciones monolíticas y basadas en microservicios?

1 punto

- ☒ Permite generar aplicaciones completas con soporte para autenticación y gestión de usuarios
- ☐ Automatiza el despliegue en entornos de producción sin intervención manual
- ☐ Proporciona una interfaz gráfica para configurar aplicaciones en tiempo real
- ☐ Incluye una base de datos incorporada para simplificar el desarrollo

¿Cuál es una desventaja de usar JHipster para desarrollar aplicaciones?

1 punto

- ☒ Posibles conflictos al regenerar o modificar un proyecto existente
- ☐ Limitación en el tipo de base de datos soportada
- ☐ No se integra con herramientas de integración continua
- ☐ No permite personalizar el sistema de autenticación

¿Qué tipos de aplicaciones se pueden crear con JHipster?

1 punto

- ☒ Aplicaciones monolíticas, basadas en microservicios y aplicaciones Gateway
- ☐ Solo aplicaciones monolíticas
- ☐ Aplicaciones basadas en microservicios, pero no aplicaciones monolíticas
- ☐ Solo aplicaciones Gateway

¿Cuál es una ventaja de las aplicaciones monolíticas generadas con JHipster?

1 punto

- ☒ El código fuente es muy bien estructurado y fácil de modificar
- ☐ Permiten escalabilidad horizontal sin problemas de infraestructura
- ☐ Requieren menos recursos de hardware y software para ejecutarse
- ☐ Tienen un menor tiempo de arranque que las aplicaciones basadas en microservicios

¿Cuál es la función del microservicio Gateway en JHipster?

1 punto

- ☒ Actuar como punto de entrada para comunicarse con un grupo de microservicios y API de terceros
- ☐ Gestionar la autenticación y autorización de usuarios
- ☐ Permitir la generación automática de código frontend
- ☐ Facilitar el despliegue de aplicaciones en múltiples entornos

¿Qué característica describe mejor a JHipster?

1 punto

- ☒ Una herramienta de scaffolding que permite generar aplicaciones web completas rápidamente
- ☐ Un framework para crear servicios web utilizando GraphQL
- ☐ Una biblioteca para desarrollar aplicaciones móviles nativas
- ☐ Un generador de scripts de despliegue para entornos Docker y Kubernetes

¿Cuál es una característica del sistema de autenticación y gestión de usuarios de JHipster?

1 punto

- ☒ Soporte para autenticación OAuth2 y OpenID Connect
- ☐ Solo permite autenticación con nombre de usuario y contraseña
- ☐ No incluye soporte para autenticación multifactor
- ☐ Requiere configuración manual para habilitar la gestión de roles y permisos

¿Qué se puede hacer con JHipster Domain Language (JDL)?

1 punto

- ☒ Describir entidades y relaciones para aplicaciones generadas con JHipster
- ☐ Crear pruebas unitarias y de integración para aplicaciones JHipster
- ☐ Configurar el despliegue automático en entornos de producción
- ☐ Definir la estructura de carpetas y archivos de un proyecto JHipster

¿Cuál es uno de los pasos para instalar JHipster en una máquina local?

1 punto

- ☒ Instalar Node.js para soporte de herramientas basadas en JavaScript
- ☐ Descargar y compilar el código fuente de JHipster
- ☐ Instalar un servidor de aplicaciones como Tomcat o WildFly
- ☐ Configurar un entorno Docker para ejecutar aplicaciones JHipster

¿Cuál es el propósito de las dependencias spring-boot-starter-web-services y wsdl4j en un proyecto JHipster? 1 punto

- ☒ Proporcionar soporte para servicios web basados en SOAP
- ☐ Permitir la conexión a bases de datos relacionales y no relacionales
- ☐ Agregar soporte para autenticación y autorización
- ☐ Habilitar la creación de endpoints RESTful

¿Cuál es la definición de "Big Data" según Gartner?

1 punto

- ☒ Datos de alto volumen, alta velocidad y alta variedad que requieren procesamiento innovador.
- ☐ Conjunto de datos estructurados que se procesan utilizando SQL.
- ☐ Información que demanda procesamiento por lotes únicamente.
- ☐ Conjunto de datos distribuidos para procesamiento en tiempo real.

¿Qué caracteriza a un sistema de Big Data?

1 punto

- ☐ Procesamiento centralizado y almacenamiento local.
- ☒ Alta capacidad de procesamiento distribuido y escalabilidad.
- ☐ Uso exclusivo de datos estructurados en SQL.
- ☐ Dependencia en arquitecturas monolíticas y tecnologías antiguas.

¿Cuál es una característica clave de Apache Spark?

1 punto

- ☐ Procesamiento en disco para mantener costos bajos.
- ☐ Enfoque en procesamiento por lotes únicamente.
- ☒ Uso de RDDs (Resilient Distributed Datasets) para procesamiento en memoria.
- ☐ Integración limitada con otros ecosistemas de Big Data.

¿Cuál es la diferencia entre un DataFrame y un RDD en Apache Spark?

1 punto

- ☒ DataFrames tienen esquema definido, mientras que RDDs no.
- ☐ DataFrames son inmutables, mientras que RDDs no lo son.
- ☐ DataFrames son para procesamiento por lotes y RDDs para streaming.
- ☐ No hay diferencias entre DataFrames y RDDs.

¿Qué es la arquitectura Lambda en Big Data?

1 punto

- ☒ Divide el procesamiento en capas para manejar tanto lotes como flujos de datos.
- ☐ Unifica el procesamiento por lotes y el streaming en una sola capa.
- ☐ Se enfoca exclusivamente en procesamiento en tiempo real.
- ☐ Utiliza microservicios para escalar de forma dinámica.

¿Qué es PySpark?

1 punto

- ☒ Una interfaz de Spark para Python.
- ☐ Un framework para desarrollar aplicaciones web con Python.
- ☐ Un módulo de Spark para ejecutar tareas en R.
- ☐ Un lenguaje de programación utilizado por Apache Spark.

¿Cuál es el propósito del RDD lineage en Apache Spark?

1 punto

- ☐ Proporcionar trazabilidad y tolerancia a fallos.
- ☐ Identificar errores de programación en tiempo real.
- ☐ Mantener una lista de transformaciones aplicadas a los datos
- ☒ Proporcionar trazabilidad y tolerancia a fallos. y mantener una lista de transformaciones aplicadas a los datos.

¿Qué es una transformación en Apache Spark?

1 punto

- ☒ Una operación que construye un nuevo RDD a partir de uno anterior.
- ☐ Una operación que calcula un resultado y lo devuelve al controlador.
- ☐ Una operación que se ejecuta de inmediato.
- ☐ Ninguna de las anteriores.

¿Cuál es la diferencia clave entre arquitectura Lambda y Kappa?

1 punto

- ☒ Lambda utiliza dos capas para procesar datos, mientras que Kappa utiliza una única capa de streaming.
- ☐ Kappa es exclusiva para microservicios, mientras que Lambda es para aplicaciones monolíticas.
- ☐ Lambda es para datos en tiempo real, mientras que Kappa es para datos históricos.
- ☐ No hay diferencia entre Lambda y Kappa.

¿Qué hace el comando `sc.textFile("file.txt")` en PySpark?

1 punto

- ☒ Crea un RDD a partir de un archivo de texto.
- ☐ Lee y almacena el contenido de un archivo de texto en memoria.
- ☐ Inicia un proceso de lectura en tiempo real.
- ☐ Ninguna de las anteriores.

¿Cuál de las siguientes afirmaciones sobre el protocolo UDP es correcta?

1 punto

- ☐ UDP garantiza la entrega de los datos en orden.
- ☐ UDP utiliza un mecanismo de control de errores y retransmisión.
- ☒ UDP no requiere establecer una conexión antes de enviar datos.
- ☐ UDP es más lento que TCP.

¿Cuál de las siguientes capas del modelo OSI es responsable del transporte de datos? 1 punto

- ☐ Capa 2
- ☐ Capa 3
- ☒ Capa 4
- ☐ Capa 5

¿Qué protocolo es más adecuado para aplicaciones donde la confiabilidad y el orden son importantes? 1 punto

- ☐ UDP
- ☒ TCP
- ☐ HTTP
- ☐ FTP

¿Qué tipo de comunicación permite WebSockets? 1 punto

- ☐ Comunicación unidireccional
- ☒ Comunicación bidireccional
- ☐ Comunicación solo para envío de archivos
- ☐ Comunicación solo para transmisión de audio

¿Qué característica diferencia a RPC (Remote Procedure Call) de otras formas de comunicación?

1 punto

- ☒ Permite la ejecución de código en otra máquina mediante una abstracción.
- ☐ Requiere una conexión persistente como TCP.
- ☐ Utiliza solo el protocolo UDP para transmisión.
- ☐ Es solo para comunicación interna entre componentes locales.

¿Cuál es la principal ventaja del uso de sockets en un sistema cliente-servidor?

1 punto

- ☒ Permiten la comunicación directa entre procesos a través de una red.
- ☐ Requieren menos recursos que otros métodos.
- ☐ Aseguran la confiabilidad y el orden de los datos.
- ☐ Solo se utilizan para conexiones locales.

¿Cuál es el uso típico del protocolo HTTP?

1 punto

- ☒ Comunicación entre aplicaciones cliente-servidor basada en solicitudes y respuestas.
- ☐ Transmisión de datos sin garantía de orden.
- ☐ Para envío de archivos grandes de forma rápida.
- ☐ Solo para comunicación segura con encriptación.

¿Cuál es una ventaja del uso de WebSockets sobre HTTP?

1 punto

- ☒ Permiten una comunicación bidireccional persistente.
- ☐ Requieren menos recursos de la red.
- ☐ Son más seguros que HTTP.
- ☐ Utilizan un mecanismo de confirmación como TCP.

¿Qué capa del modelo OSI se encarga de la presentación de datos para el usuario?

1 punto

- ☐ Capa 4
- ☐ Capa 5
- ☒ Capa 6
- ☐ Capa 7

¿Cuál de las siguientes afirmaciones sobre el protocolo TCP es correcta?

1 punto

- ☒ TCP garantiza la entrega y el orden de los paquetes.
- ☐ TCP es más rápido que UDP.
- ☐ TCP no requiere una conexión previa.
- ☐ TCP se utiliza principalmente para streaming de video.

¿Cuál es la diferencia entre WebSockets y HTTP en términos de comunicación?

1 punto

- ☒ WebSockets permiten comunicación bidireccional, mientras que HTTP es unidireccional.
- ☐ HTTP es más seguro que WebSockets.
- ☐ WebSockets requieren más recursos que HTTP.
- ☐ HTTP solo se utiliza para aplicaciones web, mientras que WebSockets se usa para otros tipos de aplicaciones.

¿Qué caracteriza a los procedimientos remotos (RPC) en la comunicación distribuida?

1 punto

- ☒ Permiten que el código en una máquina ejecute procedimientos en otra máquina.
- ☐ Requieren conexiones persistentes como TCP.
- ☐ Se utilizan principalmente para comunicación interna.
- ☐ No se pueden usar para comunicaciones de red.

¿Qué protocolo es comúnmente utilizado para el envío de correos electrónicos?

1 punto

- ☐ HTTP
- ☒ SMTP
- ☐ FTP
- ☐ UDP

¿Cuál es el propósito de la capa de transporte en el modelo OSI?

1 punto

- ☒ Proporcionar un canal de comunicación confiable entre sistemas.
- ☐ Gestionar el tráfico de red.
- ☐ Establecer conexiones físicas.
- ☐ Asegurar la presentación de datos correctos.

¿Qué aspecto del protocolo TCP lo hace más confiable que UDP?

1 punto

- ☒ TCP utiliza mecanismos para garantizar la entrega y el orden de los datos.
- ☐ TCP no requiere conexión previa.
- ☐ TCP es más rápido en la transmisión de datos.
- ☐ TCP se utiliza principalmente para aplicaciones de voz y video.

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios