



# Procesamiento en tiempo real de datos distribuidos

Guía para el profesor  
Clave: LSCA2322

## Contenido

Datos generales .....	3
Competencia global .....	3
Competencias esenciales .....	3
Introducción .....	4
Información general .....	5
Calendario de entregas semestral .....	7
Temario .....	8
Preguntas frecuentes .....	9
Recomendaciones para la explicación de temas, actividades y proyecto .....	10
Prácticas de bienestar .....	24

## Datos generales

Nombre del certificado: Procesamiento en tiempo real de datos distribuidos

Nivel: Profesional

Modalidad: Presencial

Clave: LSCA2322

## Competencia global

Aplica técnicas fundamentales de procesamiento de datos distribuidos y de aprendizaje automático utilizando Apache Spark en un entorno de *big data*, optimizando aplicaciones y flujos de trabajo para resolver problemas del mundo real.

## Competencias esenciales

- Agilidad para el aprendizaje
- Adaptabilidad
- Solución de problemas
- Colaboración
- Comunicación efectiva

## Introducción

¡Bienvenido al certificado **Procesamiento en tiempo real de datos distribuidos**! En este curso explorarás los principios y técnicas esenciales para el manejo de grandes volúmenes de datos en entornos distribuidos mediante Apache Spark, una de las herramientas más destacadas en el campo del *big data*.

Aprenderás desde la instalación y configuración básica de Spark hasta conceptos avanzados como optimización de flujos de trabajo, almacenamiento en caché, análisis de grafos y el uso de Spark MLlib y SparkR para desarrollar modelos de aprendizaje automático y minería de datos. Además, conocerás la arquitectura de datos distribuidos, las mejores prácticas de desarrollo, pruebas para maximizar el rendimiento y la eficiencia en aplicaciones reales.

Estas habilidades te ayudarán a consolidar tu perfil técnico en el procesamiento de datos masivos, y te prepararán para enfrentar desafíos en sectores como la industria, el comercio electrónico y la banca. Podrás aplicar estos conocimientos en la toma de decisiones basada en datos, en la optimización de procesos y en el desarrollo de soluciones innovadoras y escalables para posicionarte como un profesional capaz de diseñar, implementar y liderar proyectos que requieran un manejo avanzado de datos distribuidos y aprendizaje automático.

## Información general

### Metodología

Este curso contempla la lectura del contenido previo a la realización de las prácticas, actividades o avances del proyecto, según corresponda en cada uno de los temas.

En cada tema, encontrarás:

- Una introducción que presenta el contexto y la relevancia del tema a estudiar.
- La explicación, en donde se desarrollan los conceptos clave a través de definiciones, ejemplos aplicados, recursos visuales y referencias académicas.
- El cierre, que resume los aprendizajes principales e invita a reflexionar sobre su aplicación práctica.
- El checkpoint, una sección de autoevaluación que te permite verificar tu comprensión del tema mediante indicadores clave.
- Las referencias bibliográficas donde se incluyen las fuentes consultadas para sustentar el contenido presentado.

A través del curso debes trabajar en lo siguiente:

- 5 actividades
- 2 entregas del proyecto
- 1 presentación del proyecto

Además, en cada sesión de clase deberás realizar las prácticas indicadas por el profesor, las cuales están diseñadas para apoyar el desarrollo de la competencia del curso. Estas prácticas no tienen un valor en la evaluación, pero su realización es indispensable para poder realizar el proyecto.

¡Que tengas una excelente experiencia de aprendizaje en este certificado!

### Bibliografía y software

#### Bibliografía opcional

- Gowda. D. (2024). *Apache Spark for Machine Learning: Build and deploy high-performance big data AI solutions for large-scale clusters*. Reino Unido: Packt. [https://www.amazon.com.mx/Apache-Spark-Machine-Learning-high-performance-ebook/dp/B0DHCKGY2/ref=sr\\_1\\_1?\\_\\_mk\\_es\\_MX=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=3E4TZWU78Z3IL&dib=eyJ2IjojMSJ9.LehoxGMueSUGi-QdZOKbMM6zMQy1IPO5ZR\\_nksJmEw4ZiWA8lCt0k2DaJ2u0KLlIXJs2v8vfQXpui41jqy5U-x47XT0tL3VdywSHvB0feRk9Wu4jc6xDZIVTcVClmE87j0C13Cm0cUVX-4rhTDKvj-ErCaT17\\_WGI\\_OjYTQzATQ56lc41xQ-IW0GoqC4WUkOIIQUcVVCAfgfodgoinQR4wVjny4u5Nw\\_J80ZQILvkQYT7ru1axFgbcTt1Oxog4g-0wNs1B8ZF5IS2SEBWr5-NJIAZFn9adrldlognw8jQ.uOLmzOCFT9zjs2\\_P\\_ybm\\_XYGtnfJEReGAK\\_XM2WMrK4&dib\\_tag=se&keywords=machine+Learning+with+Apache+Spark&qid=1728527116&sprefix=machine+learning+with+a+apache+spark%2Caps%2C281&sr=8-1](https://www.amazon.com.mx/Apache-Spark-Machine-Learning-high-performance-ebook/dp/B0DHCKGY2/ref=sr_1_1?__mk_es_MX=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=3E4TZWU78Z3IL&dib=eyJ2IjojMSJ9.LehoxGMueSUGi-QdZOKbMM6zMQy1IPO5ZR_nksJmEw4ZiWA8lCt0k2DaJ2u0KLlIXJs2v8vfQXpui41jqy5U-x47XT0tL3VdywSHvB0feRk9Wu4jc6xDZIVTcVClmE87j0C13Cm0cUVX-4rhTDKvj-ErCaT17_WGI_OjYTQzATQ56lc41xQ-IW0GoqC4WUkOIIQUcVVCAfgfodgoinQR4wVjny4u5Nw_J80ZQILvkQYT7ru1axFgbcTt1Oxog4g-0wNs1B8ZF5IS2SEBWr5-NJIAZFn9adrldlognw8jQ.uOLmzOCFT9zjs2_P_ybm_XYGtnfJEReGAK_XM2WMrK4&dib_tag=se&keywords=machine+Learning+with+Apache+Spark&qid=1728527116&sprefix=machine+learning+with+a+apache+spark%2Caps%2C281&sr=8-1)

#### Software

- Apache Spark (incluido Spark SQL, MLlib y GraphX).

## Procesamiento en tiempo real de datos distribuidos

Guía de impartición

- Ambiente de desarrollo (Scala, Python y R).

## Evaluación

La evaluación combina los siguientes elementos:

- Actividades que abordan el contenido conceptual de los temas.
- Proyecto mediante el cual el participante demostrará que ha adquirido las habilidades y los conocimientos necesarios para acreditar el certificado. Este proyecto se divide en dos fases.
- Presentación del proyecto.

A continuación, se presenta el detalle de la evaluación:

Semana	Evaluable	Ponderación
5	Actividad I	10%
7	Actividad II	10%
9	Avance del proyecto	20%
11	Actividad III	10%
13	Actividad IV	10%
14	Certificación	10%
15	Entrega final del proyecto	20%
16	Presentación del proyecto	10%
Total		100%

## Actividades y fases del proyecto

El avance (fase I) y la entrega final del proyecto (fase II) se realizarán de manera individual.

Con el fin de fomentar el dinamismo y la interacción entre los participantes en diversos formatos, el docente alternará, durante las sesiones, intervenciones individuales, plenarios y grupales. Estas actividades enriquecerán tu perspectiva y, al mismo tiempo, te ofrecerán la oportunidad de presentar tus ideas y posturas respecto a los temas de clase.

Los resultados del avance y la entrega final del proyecto deberán presentarse a través de la plataforma tecnológica para su revisión y evaluación por parte del docente. Es muy importante que revises el esquema de evaluación y los criterios que utilizará el docente para otorgarte una calificación, con la intención de que desde

## Procesamiento en tiempo real de datos distribuidos

el inicio tengas claro el nivel de complejidad y esfuerzo que requieres para realizar las entregas semanales y garantizar tu éxito.

En caso de dudas sobre el avance, la entrega final del proyecto o el contenido, puedes contactar a tu docente a través de los medios que se te indiquen.

### Calendario de entregas semestral

Semana	Evaluable
5	Actividad I
7	Actividad II
9	Avance del proyecto
11	Actividad III
13	Actividad IV
14	Certificación
15	Entrega final del proyecto
16	Presentación del proyecto

## Temario

Tema 1. Introducción a Apache Spark

Tema 2. Arquitectura de Spark y Resilient Distributed Datasets (RDD)

Tema 3. Transformaciones y acciones en Spark

Tema 4. Optimización de operaciones en Spark

Tema 5. *Caching* y persistencia de datos en Spark

Tema 6. Serialización de datos en Spark

Tema 7. Desarrollo y pruebas de aplicaciones Spark

Tema 8. Introducción a Spark SQL y DataFrames

Tema 9. Spark MLlib: fundamentos de *Machine Learning*

Tema 10. Árboles de decisión y *random forest* en Spark MLlib

Tema 11. *Clustering* y análisis de datos no supervisados en MLlib

Tema 12. Análisis de grafos con Spark GraphX

Tema 13. Introducción a SparkR

Tema 14. Machine Learning en SparkR

## Preguntas frecuentes

### ¿En dónde o a quién le reporto un error detectado en el contenido?

Lo puedes reportar a través del botón “Mejora tu curso”, también puedes compartir sugerencias para el contenido y actividades del certificado.

### ¿Quién me informa sobre la cantidad de sesiones y el tiempo de cada sesión cada semana?

El coordinador docente te debe proporcionar esta información.

### ¿Tengo que capturar las calificaciones en Banner y en la plataforma educativa?

Sí, es importante que captures las calificaciones en la plataforma para que los participantes estén informados de su avance y reciban retroalimentación de parte tuya sobre todo lo que realizan en esta experiencia educativa. En Banner, se lleva el registro oficial de las calificaciones de los participantes.

## Recomendaciones para la explicación de temas, actividades y proyecto

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 1:

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia contextualizando el procesamiento de grandes volúmenes de datos en escenarios reales, como en instituciones financieras, donde la rapidez en el análisis es crucial para la toma de decisiones. Este enfoque práctico facilitará que los estudiantes comprendan la utilidad de Spark más allá del marco teórico.
- Presenta el ecosistema de Apache Spark a través de sus componentes clave (Spark Core, SQL, MLlib, Streaming, GraphX), explica su función dentro de un flujo de trabajo de datos.
- La parte más propensa a generar dudas es la arquitectura maestro-esclavo y la interacción entre componentes como el driver program, executors y worker nodes. Aclara estos conceptos usando diagramas interactivos y simulaciones prácticas en PySpark, de forma que los estudiantes visualicen cómo se distribuyen y ejecutan las tareas en un clúster.
- Explica con mayor detalle la diferencia entre procesamiento en memoria y en disco, el funcionamiento de Spark SQL y DataFrames en combinación con Spark Streaming, y la integración de MLlib en *pipelines* de machine learning.
- Dedicar tiempo a comparar Spark con Hadoop para consolidar el entendimiento de cuándo y por qué usar Apache Spark en proyectos reales.
- Para lograr un aprendizaje significativo, es esencial que los estudiantes interactúen con la herramienta desde el entorno de Google Colab, instalando y configurando PySpark paso a paso. Así se fomenta una comprensión técnica mientras se mantiene una conexión constante con su aplicación práctica.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 2:

Al profesor impartidor, se le recomienda lo siguiente:

- Comienza dando contexto sobre la necesidad del procesamiento distribuido de datos en sectores críticos, como la banca, donde millones de transacciones deben ser evaluadas en tiempo real.
- Menciona que los RDD son la base sobre la que se construye el procesamiento en Apache Spark. Introduce el concepto con ejemplos visuales de cómo los datos se dividen, distribuyen y procesan en paralelo en un clúster.
- Incluye ejercicios prácticos en Google Colab para que los estudiantes configuren Spark, creen RDD, apliquen transformaciones y acciones, y observen cómo se gestionan los datos en particiones. Este enfoque les permitirá comprender con claridad los principios de paralelismo, resiliencia y eficiencia del sistema.
- Las dudas más frecuentes pueden surgir en torno al concepto de linaje, la evaluación perezosa y la tolerancia a fallos. Para clarificar estas ideas, apóyate en la herramienta `toDebugString()` para mostrar el grafo de transformación de un RDD y simular fallos controlados en un clúster.

- Explica con especial detalle los conceptos de particionamiento, reconstrucción de datos y diferencias entre RDD base y derivados.
- Enfatiza cómo el uso estratégico de los RDD resuelve problemas reales, por ejemplo, la detección de fraudes o la segmentación de clientes, y cómo su diseño inmutable y distribuido se adapta a las exigencias de la ciencia de datos moderna.

### **Notas para el profesor impartidor. Estas corresponden a la explicación del tema 3:**

Al profesor impartidor, se le recomienda lo siguiente:

- Comienza explicando cómo Spark estructura sus operaciones mediante un modelo de ejecución basado en transformaciones perezosas y acciones. Inicia con un caso real, por ejemplo, el uso de Spark en plataformas como Netflix o Uber, que procesan datos en tiempo real para generar recomendaciones o mejorar rutas.
- Construye un ejemplo sencillo paso a paso, por ejemplo, filtrar pedidos o calcular ingresos de productos, resalta cómo las transformaciones (como map o filter) generan nuevos RDD sin ejecutar inmediatamente, y cómo una acción (como count o reduce) desencadena todo el procesamiento acumulado.
- Introduce de manera gradual las ideas de grafo de linaje, persistencia y particiones mediante visualizaciones y código interactivo en Colab, para reforzar su aplicación práctica.
- Explica detalladamente cuándo conviene usar cache() o persist(), y cómo elegir entre repartition() y coalesce() en función del tamaño de los datos y la etapa del flujo de trabajo. Enfatiza que entender estas diferencias y aplicarlas de manera correcta mejora el rendimiento en clústeres distribuidos y permite diseñar flujos de trabajo eficientes, escalables y resilientes para entornos de big data.
- Los puntos que pueden generar más dudas son la evaluación perezosa y la optimización del grafo de linaje, ya que no son comportamientos evidentes sin ver su efecto en ejecución. Para aclararlos, muestra ejecuciones diferidas de transformaciones que no se concretan hasta invocar una acción.

### **Notas para el profesor impartidor. Estas corresponden a la explicación del tema 4:**

Al profesor impartidor, se le recomienda lo siguiente:

- Guía a los estudiantes desde la comprensión de las transformaciones perezosas hasta el análisis del impacto del *shuffling*, la persistencia de datos y el uso eficiente del particionamiento.
- Inicia con un ejemplo comparativo entre groupByKey y reduceByKey, mostrando cómo pequeñas decisiones influyen en el rendimiento. Luego, introduce el concepto de DAG (grafo acíclico dirigido) para explicar la evaluación diferida y cómo Spark optimiza automáticamente el plan de ejecución.
- Aborda técnicas prácticas como cache(), persist() y los métodos repartition() y coalesce() con ejemplos en Google Colab, que permitan a los estudiantes observar cómo se distribuyen los datos entre particiones y cómo estos ajustes afectan el uso del CPU y memoria.
- El punto más confuso para los estudiantes suele en qué momento aplicar repartition() o coalesce(), y cómo el particionamiento afecta el rendimiento en operaciones con join o reduceByKey. Para resolver estas dudas combina teoría con experimentación práctica: usar .getNumPartitions() y .glom().collect()

para visualizar la distribución de los datos y modificar parámetros como `spark.sql.shuffle.partitions` para observar los cambios.

- Los conceptos que requieren mayor detalle son el particionamiento por clave, los parámetros del ejecutor (`spark.executor.memory`, `spark.executor.cores`) y las consecuencias del *shuffle*, ya que son críticos para tareas distribuidas y afectan la eficiencia en clústeres. Subraya que el dominio de estas optimizaciones permite construir flujos de trabajo escalables y sostenibles para aplicaciones en big data reales.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 5:

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia el tema a partir de la necesidad de evitar recalculando los mismos datos múltiples veces en entornos distribuidos. Es clave que expliques que Spark, por su naturaleza de evaluación perezosa, solo ejecuta transformaciones cuando una acción las activa. Esto permite optimizar el uso de memoria y procesamiento a través de técnicas como `cache()` y `persist()`.
- Centra el enfoque inicial en identificar cuándo tiene sentido almacenar datos en memoria (caching) y cuándo se requiere mayor control sobre el almacenamiento (persistencia). Para ello, emplea ejemplos con RDD y DataFrames, resaltando el uso de niveles como `MEMORY_AND_DISK`, `DISK_ONLY` y `MEMORY_ONLY_SER`, y refuerza la comprensión con ejercicios en Google Colab.
- Se recomienda utilizar casos cercanos al estudiante, como ventas por región, para contextualizar la eficiencia en el uso de recursos.
- Las dudas más frecuentes se enfocan en la selección adecuada del nivel de almacenamiento y la diferencia entre `cache()` y `persist()`, así como en el momento oportuno para liberar recursos mediante `unpersist()`. Para resolver estas inquietudes, apóyate en visualizaciones mediante Spark UI (habilitable en Colab con `ngrok`), permitiendo que el estudiante observe en tiempo real el uso de memoria y disco.
- Explica detalladamente los niveles de persistencia disponibles en `StorageLevel`, cuándo usarlos y cómo combinarlos con pipelines complejos. La comprensión profunda de estos conceptos permite diseñar flujos de trabajo eficientes y escalables, minimizando costos computacionales y asegurando estabilidad operativa en ambientes de procesamiento masivo.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 6:

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia explicando la necesidad de este proceso en entornos distribuidos, destaca que Spark requiere transmitir datos entre nodos del clúster y, por lo tanto, convertir los objetos en un formato serializado es importante.
- Compara los dos principales mecanismos: **Java Serialization**, valorado por su simplicidad y uso inmediato, y **Kryo Serialization**, recomendado por su rendimiento y eficiencia en memoria. Utiliza ejemplos prácticos en Google Colab para ilustrar cómo ambos mecanismos afectan operaciones comunes como filtrado, agrupación y estadísticas sobre grandes volúmenes de datos.

- Muestra paso a paso la configuración explícita de Kryo, refuerza su aplicación en contextos donde el rendimiento es crítico.
- Explica en profundidad los siguientes conceptos: qué es la serialización, por qué influye en el rendimiento, y cómo seleccionar entre Java y Kryo según el tipo de aplicación (pequeña, prototipo o producción).
- Destaca el uso de herramientas como **Spark UI** para analizar el comportamiento de las tareas y validar empíricamente las decisiones de configuración. Así, el estudiante comprenderá la teoría y su implicación práctica en proyectos reales.
- La parte que puede generar mayor confusión es la necesidad de **registrar clases en Kryo** y cómo este paso mejora aún más el rendimiento. Para resolver estas dudas, ejemplifica cuándo es necesario ese registro y cómo hacerlo de manera correcta, subrayando los beneficios medibles en tiempo y consumo de recursos.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 7:

Al profesor impartidor, se le recomienda lo siguiente:

- Guía a los estudiantes a través del diseño eficiente de soluciones distribuidas utilizando buenas prácticas de codificación con PySpark o Scala.
- Explica la instalación y configuración del entorno en Google Colab, seguido por el uso estratégico de estructuras de datos como RDD y DataFrames, enfatizando la selección adecuada según el caso de uso.
- Subraya la importancia del particionamiento, la persistencia selectiva de datos y técnicas de optimización como *broadcast joins* y Pandas UDFs. Una forma efectiva de abordar el tema es mediante ejemplos progresivos que demuestren cómo estas decisiones afectan el rendimiento.
- Integra de manera práctica el uso de `explain()`, `cache()`, `unpersist()` y la herramienta Spark UI durante la sesión para reforzar el análisis del rendimiento.
- Explica detalladamente los conceptos de shuffling, particiones, plan de ejecución físico y el uso de funciones como `collect()` y `take()`. Además, profundiza en los conceptos de pruebas unitarias, de integración y de rendimiento, mostrando cómo simular conjuntos de datos realistas y cómo interpretar métricas para mejorar la escalabilidad. Con este enfoque, los estudiantes estarán mejor preparados para desarrollar, probar y escalar aplicaciones Spark de manera robusta y eficiente.
- La sección más compleja son las **pruebas y depuración en entornos distribuidos**, ya que requiere la comprensión de cómo se propagan los errores en un clúster y cómo interpretar la ejecución distribuida. Para resolver esto, apóyate en pruebas prácticas que combinen funciones personalizadas, validaciones con `assert`, y visualización de cuellos de botella mediante Spark UI.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 8:

Al profesor impartidor, se le recomienda lo siguiente:

- Enfatiza la evolución desde los RDD hacia estructuras optimizadas como los DataFrames.

- Explica la necesidad de una abstracción más eficiente para datos estructurados, introduce las ventajas que ofrecen los DataFrames en términos de rendimiento, facilidad de uso y compatibilidad con SQL.
- Utiliza ejemplos concretos del entorno bancario o de comercio electrónico para contextualizar su aplicación práctica. Luego, enseña el uso de Spark SQL como interfaz declarativa para consultar datos con lenguaje SQL tradicional, ilustra cómo registrar vistas temporales o persistentes, aplicar filtros y realizar agregaciones.
- El tema en el cual surgen dudas es la **optimización del rendimiento con Catalyst, Tungsten y Adaptive Query Execution (AQE)**, ya que estos conceptos operan a nivel interno del motor de ejecución. Para aclarar estas inquietudes, presenta visualizaciones del plan lógico y físico de una consulta usando explain(), y muestra cómo se transforma y optimiza automáticamente. Enfatiza la diferencia entre funciones nativas y UDFs, destacando que estas últimas no son optimizadas por Catalyst.
- Explica de manera detallada cómo crear y aplicar UDFs tanto en la API de DataFrames como en consultas SQL, y refuerza los conceptos mediante prácticas en Google Colab conectando fuentes externas con JDBC. Con este enfoque, los estudiantes entenderán cómo usar Spark SQL para sacarle el máximo provecho en el análisis eficiente de datos estructurados en entornos distribuidos.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 9:

Al profesor impartidor, se le recomienda lo siguiente:

- Aborda el tema de Spark MLlib desde una perspectiva aplicada, explica cómo Apache Spark facilita el desarrollo de modelos de machine learning en entornos distribuidos.
- Inicia mostrando casos reales de uso como los sistemas de recomendación en Netflix o Spotify, contextualiza la relevancia del aprendizaje automático en aplicaciones a gran escala. Luego, introduce la arquitectura y estructuras de datos básicas como DataFrame, Vector y LabeledPoint, destaca su papel en la construcción y entrenamiento de modelos.
- Para facilitar el aprendizaje desarrolla ejercicios prácticos con regresión logística o clustering con K-Means, y compara el enfoque supervisado versus no supervisado. Utiliza código explicativo y visualizaciones del flujo de trabajo completo: desde la preparación de los datos hasta la evaluación del modelo.
- Al finalizar el tema, los estudiantes deben ser capaces de implementar un flujo de trabajo de machine learning con MLlib, seleccionar el modelo adecuado según el tipo de datos y contexto, y comprender cómo desplegarlo de forma escalable, incluyendo entornos como Kubernetes o Structured Streaming para casos en tiempo real.
- El tema que causa mayor confusión es la construcción de vectores de características y la integración con fuentes externas como bases de datos SQL. Por ello, explica de manera detallada el uso de VectorAssembler, StringIndexer y la transformación de columnas categóricas, incluye ejemplos de conexión con bases de datos usando JDBC. Explica a fondo las diferencias entre vectores densos y dispersos, así como la utilidad de herramientas como Delta Lake para el control de versiones y la trazabilidad de modelos.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 10:

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia con un caso práctico que motive al grupo, por ejemplo, la evaluación de riesgo crediticio, y muestra cómo un árbol de decisión puede replicar razonamientos humanos al dividir datos en pasos lógicos y comprensibles.
- Explica cada componente del árbol (nodo raíz, nodos internos, ramas y hojas), complementa la teoría con diagramas jerárquicos. La implementación con Spark MLlib debe centrarse en la construcción del flujo completo: creación del DataFrame, ensamblaje de características con VectorAssembler, entrenamiento con DecisionTreeClassifier, y predicción.
- Destaca la configuración de parámetros como maxDepth o impurity, ya que son clave para evitar el sobreajuste y mejorar la generalización del modelo.
- Profundiza en temas como la importancia de variables, la evaluación con métricas AUC, y el uso del modelo en sistemas en tiempo real, lo cual permite consolidar el conocimiento aplicado en escenarios reales de gran escala, como FinTech o e-commerce.
- La parte más compleja del tema es la diferencia conceptual y práctica entre un árbol de decisión individual y un random forest. Enfatiza el principio de ensamblaje (*bagging*), donde múltiples árboles se entrenan sobre subconjuntos de datos y variables, y sus predicciones se combinan para obtener resultados más precisos y robustos. Explica visualmente este proceso, y ejemplifica la implementación en Spark MLlib con RandomForestClassifier, compara los resultados con modelos individuales.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 11:

Al profesor impartidor, se le recomienda lo siguiente:

- Explica el enfoque del aprendizaje no supervisado, destaca que se utiliza cuando no se cuenta con etiquetas previas. Contextualiza el tema con un caso real, como la detección de lavado de dinero, para que los estudiantes comprendan la manera en que los algoritmos de clustering como K-Means permiten identificar patrones anómalos sin necesidad de ejemplos previos.
- Posteriormente, guía paso a paso la implementación de un modelo en Apache Spark MLlib, desde la creación del DataFrame hasta la predicción de los clústeres.
- Enfatiza el uso de VectorAssembler, los parámetros del modelo (k, initMode, tol, maxIter) y el análisis de resultados con predictionCol.
- Complementa la construcción del modelo con ejemplos gráficos del método del codo y la métrica *Silhouette* para validar la calidad de los grupos generados.
- Diferencia los tres algoritmos disponibles en MLlib (K-Means, PIC, LDA), resalta su aplicabilidad y parámetros más relevantes: en K-Means, la elección del valor k y el método de inicialización; en PIC, la similitud entre entidades; y en LDA, el análisis temático de textos. Refuerza estos conceptos con ejemplos del sector financiero o comercial, donde se analicen perfiles de clientes, transacciones o artículos.

- El tema donde surgen más dudas es la interpretación de los resultados del clustering, en especial en cómo determinar el número óptimo de clústeres y cuándo un grupo representa un comportamiento atípico. Para aclararlo, explica con detalle el cálculo de la métrica Silhouette y realiza simulaciones donde se visualicen casos mal agrupados.

### **Notas para el profesor impartidor. Estas corresponden a la explicación del tema 12:**

Al profesor impartidor, se le recomienda lo siguiente:

- Comienza contextualizando el análisis de grafos como una técnica poderosa para representar relaciones complejas en dominios, por ejemplo, de redes sociales y detección de fraude o logística.
- Usa un caso como el de Laura que trabaja en un banco, para introducir el problema: detectar conexiones ocultas entre clientes aparentemente independientes. A partir de ahí, explica los conceptos fundamentales de teoría de grafos –nodos, aristas, grafos dirigidos y multigrafos– y su implementación en Spark GraphX, destacando su capacidad para representar relaciones con propiedades tanto en vértices como en aristas.
- Durante la clase, trabaja un ejemplo práctico en Scala, donde se construya un grafo con personas y relaciones, aplicando funciones como `mapVertices`, `subgraph`, y `joinVertices` para que los estudiantes entiendan la manera de transformar estructuras de red.
- Asimismo, explica con detalle el uso de triplets y su utilidad para combinar información de nodos y aristas. Profundiza conceptos como los operadores `reverse`, `joinVertices`, `outerJoinVertices`, en especial Pregel, ya que este último es clave para resolver problemas de caminos mínimos o flujos óptimos en redes financieras. Un análisis visual de los grafos y su interpretación contextual contribuirá a que los estudiantes comprendan mejor su aplicabilidad en problemas reales.
- Las mayores dudas aparecen al aplicar algoritmos iterativos como PageRank, Pregel o `aggregateMessages`, ya que implican múltiples pasos lógicos y la comprensión de mensajes entre nodos. Para aclarar esto, presenta visualmente cómo fluye la información entre los nodos, acompañando el código con explicaciones paso a paso del resultado esperado.

### **Notas para el profesor impartidor. Estas corresponden a la explicación del tema 13:**

Al profesor impartidor, se le recomienda lo siguiente:

- Explica el contexto actual del análisis de datos en la era del big data, enfatizando que R, aunque poderoso, se ve limitado cuando los volúmenes de datos superan la capacidad de la memoria local.
- Introduce el caso de Ana Morales como ejemplo práctico para generar empatía con los estudiantes que usan R en su jornada diaria. A partir de este caso explica cómo SparkR permite usar el entorno y sintaxis de R, pero con el poder de procesamiento distribuido de Apache Spark.
- Muestra de manera visual el flujo de integración entre R, Java y Spark, así como una demostración paso a paso de la instalación en Google Colab, destacando que no requiere configuración local compleja, lo cual es ideal para entornos educativos o de capacitación.
- Las dudas más frecuentes se presentan al diferenciar entre `data.frame`, `SparkDataFrame` y los `DataFrames` de PySpark, ya que tienen nombres similares, pero comportamientos distintos. Para aclarar esto, utiliza una tabla comparativa y muestra ejemplos en tiempo real con las funciones `groupBy()` y `summarize()`.

- También suele haber confusión con los errores técnicos comunes, como la falta de Java o la no detección de SparkR; por ello, dedica tiempo a explicar las rutas, variables de entorno y funciones como `.libPaths()`.
- Profundiza en los conceptos de evaluación diferida (*lazy evaluation*) y cómo se ejecutan los procesos en los nodos distribuidos, ya que entender la arquitectura detrás de SparkR permitirá a los alumnos usar esta herramienta de forma más efectiva y con mayor criterio técnico.

### Notas para el profesor impartidor. Estas corresponden a la explicación del tema 14:

Al profesor impartidor, se le recomienda lo siguiente:

- Explica la diferencia entre regresión y clasificación dentro del contexto del aprendizaje supervisado, vincula estas técnicas con casos reales como los enfrentados por Ana Morales en su trabajo.
- Comienza con una revisión rápida del flujo de un modelo predictivo (selección de variables, entrenamiento, evaluación y predicción), para después abordar el uso práctico de `spark.glm()` en modelos de regresión lineal y logística.
- Durante la sesión, enfatiza cómo SparkR permite escalar los modelos de machine learning a volúmenes masivos de datos, algo que no es posible con R tradicional.
- Facilita ejercicios prácticos en Google Colab, mostrando paso a paso cómo generar datos simulados, entrenar modelos, interpretar coeficientes y validar resultados con visualizaciones (como dispersión, curva ROC y matriz de confusión).
- Las dudas más frecuentes se concentran en tres áreas: (1) la diferencia entre una predicción numérica y una clasificación binaria; (2) la interpretación de los coeficientes de los modelos y su significancia estadística; y (3) la correcta evaluación del rendimiento del modelo. Para aclararlas, usa ejemplos numéricos simples, reforzados con gráficas visuales y explicaciones del significado de métricas como AUC, precisión, deviance y  $R^2$ .
- Dedicar tiempo a explicar las funciones `predict()`, `collect()` y cómo transformar las probabilidades predichas en clases mediante umbrales. Además, aclara los criterios para seleccionar un tipo de modelo según el tipo de variable dependiente. Por último, describe de manera más detallada la curva ROC y el concepto de sobreajuste, ya que son fundamentales para que los alumnos comprendan el valor y los límites de los modelos construidos en SparkR.

### Notas para el profesor impartidor. Estas corresponden a la explicación de la actividad I:

Al profesor impartidor, se le recomienda lo siguiente:

- Explica los conceptos fundamentales de Apache Spark y RDD, enfócate en su arquitectura, procesamiento distribuido y las ventajas frente a otras herramientas de análisis de datos.
- Contextualiza el caso práctico en el sector financiero, haciendo énfasis en cómo el análisis de datos puede prevenir fraudes y mejorar procesos bancarios.

- Guía paso a paso el entorno de configuración en Google Colab, y verifica que todos los estudiantes logren instalar Spark y cargar los datos simulados de manera correcta. Esta fase debe realizarse de forma colaborativa o supervisada, ya que problemas técnicos podrían frenar el desarrollo del resto de la actividad.
- Presta especial atención al bloque sobre la "tolerancia a fallos", ya que es ahí donde surgen más dudas. Este concepto resulta abstracto si no se acompaña de una demostración práctica, por lo que se sugiere realizar la simulación en tiempo real, interrumpiendo y reiniciando el proceso frente al grupo para explicar el linaje de los RDD y cómo Spark reconstruye los datos. Para aclarar este tema, haz un análisis visual del grafo de linaje y discútanlo grupalmente.
- Los puntos que deben explicarse con mayor detalle son la configuración de Spark en Colab, creación y manipulación de RDD (incluyendo transformaciones y acciones), y el linaje de datos como mecanismo de recuperación ante fallos. Todo esto debe complementarse con prácticas constantes para reforzar la comprensión teórica y técnica.

### **Notas para el profesor impartidor. Estas corresponden a la explicación de la actividad II:**

Al profesor impartidor, se le recomienda lo siguiente:

- Guía al estudiante en el entendimiento y aplicación del procesamiento de datos distribuidos en tiempo real con Apache Spark.
- Comienza contextualizando el problema que enfrenta la empresa Finanzas Digitales MX, lo que permite al estudiante conectar el aprendizaje técnico con un caso del mundo real.
- Asegura que los alumnos comprendan bien las herramientas de configuración necesarias en Google Colab (como PySpark y findspark), y que verifiquen que el entorno esté correctamente preparado antes de ejecutar las transformaciones.
- La creación del conjunto de datos y su conversión en RDD debe realizarse paso a paso, permitiendo que el estudiante se familiarice con las estructuras y operaciones más comunes de Spark.
- Una parte crítica del tema donde surgen dudas es el uso de las técnicas de optimización, en especial `cache()` y `partitionBy()`, así como la evaluación del rendimiento antes y después de su aplicación. Para aclarar estas dudas, realiza comparaciones en vivo con ejemplos cronometrados usando `%time`, explica el impacto de estas técnicas en la eficiencia del sistema.
- Explica detalladamente el grafo de linaje generado con `toDebugString()` en la simulación de fallo, ya que es un concepto abstracto pero clave para entender la tolerancia a fallos en Spark.
- Los conceptos que deben abordarse con mayor profundidad son: `reduceByKey`, `cache()`, `partitionBy()`, linaje de RDDs y cómo Spark evita la pérdida de datos mediante su reconstrucción lógica. Refuerza estos conceptos con ejercicios prácticos y visualizaciones que ayuden a consolidar el aprendizaje.

### **Notas para el profesor impartidor. Estas corresponden a la explicación de la actividad III:**

Al profesor impartidor, se le recomienda lo siguiente:

## **Procesamiento en tiempo real de datos distribuidos**

- Enfoca la enseñanza en la integración de Spark SQL y DataFrames dentro de un flujo analítico optimizado, destacando su aplicación en el contexto financiero.
- Revisa la configuración del entorno en Google Colab, asegurándote de que los estudiantes comprendan los parámetros habilitados para Catalyst, Tungsten y AQE, ya que son clave para la optimización automática de consultas. Una estrategia efectiva es mostrar en vivo cómo estos motores mejoran el rendimiento al comparar operaciones con y sin optimización.
- Explica el uso de consultas SQL en combinación con la API de PySpark con ejemplos paralelos que permitan evidenciar las diferencias de sintaxis, rendimiento y claridad.
- Las áreas donde surgen mayores dudas son la aplicación de broadcast join y las funciones definidas por el usuario mediante Pandas UDFs. Estos elementos requieren una comprensión intermedia de la eficiencia de los joins en entornos distribuidos y del manejo de funciones vectorizadas. Ilustra cuándo conviene usar un broadcast join (verificando el tamaño del DataFrame pequeño) y realiza ejercicios guiados para la creación y uso de Pandas UDFs, destacando su ventaja en operaciones que requieren transformación compleja por fila.
- Los conceptos que deben explicarse en profundidad son: el funcionamiento de Catalyst y Tungsten, la persistencia con `.cache()`, la repartición eficiente con `.repartition()`, el uso de vistas temporales para consultas SQL, y la integración de UDFs. Esto permitirá que los estudiantes comprendan cómo construir soluciones escalables y rápidas que respondan a las necesidades analíticas del sector financiero.

#### **Notas para el profesor impartidor. Estas corresponden a la explicación de la actividad IV:**

Al profesor impartidor, se le recomienda lo siguiente:

- Para abordar esta cuarta actividad, introduce al estudiante en el uso de técnicas de aprendizaje supervisado dentro del entorno de Apache Spark MLlib, con especial enfoque en modelos de clasificación como los árboles de decisión y random forest.
- Comienza con una explicación clara de los conceptos fundamentales de clasificación binaria, importancia del balance de clases y cómo se representa un problema de riesgo crediticio en un conjunto de datos.
- Guía paso a paso la creación del dataset simulado y su posterior carga con SparkSession y VectorAssembler, aclarando cómo se preparan los datos para que los modelos puedan procesarlos correctamente. Este acompañamiento es clave para evitar errores comunes en la transformación de características.
- Las dudas más frecuentes surgen al ajustar los hiperparámetros del modelo y al interpretar los resultados obtenidos de cada algoritmo. Por lo tanto, explica con detalle el significado de parámetros como `maxDepth`, `impurity`, `numTrees` y la manera en que estos influyen en el rendimiento del modelo y en la posibilidad de sobreajuste. Para aclarar estas dudas, compara visualmente los resultados de ambos modelos aplicando una misma entrada de prueba y explicando con métricas simples (como precisión y recall) cuál modelo generaliza mejor.

- Profundiza en los siguientes conceptos: creación y uso de `VectorAssembler`, entrenamiento y evaluación de `DecisionTreeClassifier` y `RandomForestClassifier`, interpretación del árbol con `toDebugString`, y análisis de `featureImportances`.
- Fomenta una reflexión sobre la utilidad real de estas herramientas en el sector financiero, motivando al estudiante a pensar en soluciones escalables y automatizadas para la toma de decisiones crediticias.

### **Notas para el profesor impartidor. Estas corresponden a la explicación de la certificación:**

Al profesor impartidor, se le recomienda lo siguiente:

- Para impartir esta actividad de certificación enfoca tu acompañamiento en motivar y guiar al estudiante hacia la consolidación de los conocimientos adquiridos en el curso. Inicia explicando el valor profesional y académico de la certificación "Spark Fundamentals", enfatizando cómo valida competencias clave en el manejo de Apache Spark para procesamiento de datos a gran escala.
- Orienta al estudiante en la navegación por la plataforma Cognitive Class, verificando que se registre de manera adecuada y que identifique cada uno de los módulos que debe completar. Programa espacios de consulta para resolver dudas técnicas o conceptuales conforme los estudiantes avancen en el contenido.
- El punto más susceptible a generar confusión es la realización correcta de las actividades prácticas y evaluaciones de cada módulo, en especial si se presentan errores de entorno o dificultad en el uso de herramientas específicas como `MLlib` o `GraphX`. Para atender esto, se sugiere que fomentes el trabajo autónomo acompañado de seguimiento puntual mediante retroalimentaciones individuales o foros de dudas.
- Simula una práctica de certificación para familiarizar a los estudiantes con el formato de preguntas. Los conceptos que más deben reforzarse son: arquitectura de Spark, RDD vs. `DataFrames`, uso de Spark SQL, introducción a machine learning con `MLlib` y la lógica de grafos en `GraphX`, ya que estos son pilares de la certificación y evalúan tanto teoría como habilidades prácticas.

### **Notas para el profesor impartidor. Estas corresponden a la explicación del avance del proyecto (Fase I):**

Al profesor impartidor, se le recomienda lo siguiente:

- Para abordar la Fase I del proyecto, enfócate en sentar las bases conceptuales y prácticas del uso de Apache Spark en entornos distribuidos, iniciando con la arquitectura y el modelo de programación de los RDD.
- Realiza una introducción clara del entorno de trabajo en Google Colab, guiando paso a paso en la instalación, configuración e importación de bibliotecas necesarias, en especial para quienes aún no dominan el uso de Spark en la nube.
- El acompañamiento en las primeras operaciones con RDD y `DataFrames` es crucial, ya que los errores de inicialización y comprensión de las acciones diferidas son comunes. Incentiva el análisis de preguntas reflexivas después de cada paso, promoviendo una comprensión activa del porqué detrás de cada instrucción.

- Los puntos que podrían generar más dudas en esta fase son las diferencias entre transformaciones y acciones, la visualización del linaje con `toDebugString`, el uso de `cache()` y `persist()`, y los efectos del particionamiento y la serialización. Para aclarar estos conceptos, compara de manera explícita los tiempos de ejecución antes y después de aplicar optimizaciones, mostrando con ejemplos concretos cómo impactan en el rendimiento. También es útil que hagas una demostración comparativa entre Java Serialization y Kryo, facilitando gráficamente el análisis de resultados.
- Los conceptos que deben abordarse con mayor profundidad incluyen: RDD vs DataFrame, diferimiento en la ejecución, caché vs persistencia, niveles de persistencia, y el rol de la serialización. Estas explicaciones deben complementarse con experimentación práctica y discusión grupal para fortalecer la competencia en el uso eficiente de Spark desde el inicio del proyecto.

**Notas para el profesor impartidor. Estas corresponden a la explicación de la entrega final del proyecto (Fase II):**

Al profesor impartidor, se le recomienda lo siguiente:

- Para abordar la Fase II del proyecto, enfoca la sesión en integrar herramientas avanzadas del ecosistema de Apache Spark con énfasis en su aplicación práctica para resolver problemas reales de negocio.
- Presenta cómo Spark SQL, MLlib y GraphX permiten construir flujos completos de análisis de datos, desde la consulta estructurada hasta la modelación predictiva y el análisis de relaciones complejas.
- Organiza la enseñanza en bloques prácticos por cada tecnología, iniciando por Spark SQL, avanzando con aprendizaje automático supervisado/no supervisado y culminando con el análisis de grafos. Cada sección debe incluir ejemplos guiados, sesiones de codificación activa y discusión de resultados.
- Las áreas donde surgen más dudas son la preparación de datos para MLlib (`VectorAssembler`, `LabeledPoint`), la evaluación de modelos con métricas como MAE y RMSE, la interpretación de centroides en K-Means y la utilidad práctica de GraphX. Para aclarar estas dudas, refuerza los conceptos de ingeniería de características, selección de hiperparámetros y validación de modelos.
- Explica con ejemplos visuales cómo funciona el PageRank en grafos, relacionándolo con problemas como detección de productos más influyentes.
- Los conceptos que deben explicarse con mayor profundidad incluyen: creación de DataFrames temporales para SQL, el flujo completo de un pipeline en MLlib, criterios de segmentación con K-Means, y la estructura de vértices y aristas en GraphX. Esto permitirá a los estudiantes generar soluciones avanzadas y escalables en un entorno distribuido, afianzando sus competencias técnicas con enfoque estratégico.

**Rúbrica avance del proyecto (Fase I)**

Nivel de desempeño				
Criterios de evaluación	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	%
1. Configuración funcional de Spark en Colab.	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	<b>20</b>
	Realiza una instalación y configuración básica de Apache Spark.	Muestra problemas para realizar la configuración correcta de Apache Spark.	No realiza la configuración correcta.	
2. Manipulación de RDD y comprensión de linaje y particionamiento.	25 – 23 puntos	22 – 20 puntos	19 – 0 puntos	<b>25</b>
	Manipula de manera correcta el RDD y comprende el linaje y el particionamiento.	Manipula de manera correcta el RDD, pero no logra comprender el linaje y ni el particionamiento.	No logra manipular el RDD ni comprende el linaje y particionamiento.	
3. Uso de transformaciones y acciones en RDD y DataFrames.	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	<b>20</b>
	Usa de manera adecuada las transformaciones y acciones en RDD y DataFrames.	Presenta problemas para usar de manera adecuada las transformaciones y acciones en RDD y DataFrames.	Uso incorrecto de las transformaciones y acciones en RDD y DataFrames.	
4. Aplicación de técnicas de optimización.	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	<b>20</b>
	Aplica de manera correcta las técnicas de optimización (.cache(), .repartition(), .persist()).	Presenta dificultades para aplicar las técnicas de optimización.	No aplica de manera adecuada las técnicas de optimización.	
5. Evaluación comparativa de serialización Java vs. Kryo.	15 – 13 puntos	12 – 9 puntos	8 – 0 puntos	<b>15</b>
	Realiza una evaluación comparativa correcta sobre la serialización Java vs Kryo.	Compara de manera vaga la serialización Java vs Kryo.	Realiza una evaluación comparativa incorrecta.	
<b>TOTAL</b>				<b>100%</b>

**Rúbrica de la entrega final del proyecto (Fase II)**

Criterios de evaluación	Nivel de desempeño			%
	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	
1. Implementación de prácticas de desarrollo.	15 – 13 puntos	12 – 10 puntos	9 – 0 puntos	15
	Implementa de manera correcta las prácticas de desarrollo y valida los datos en PySpark.	Implementa las prácticas de desarrollo, pero no logra validar los datos en PySpark.	No implementa las prácticas de desarrollo ni valida los datos en PySpark.	
2. Uso de Spark SQL.	15 – 13 puntos	12 – 10 puntos	9 – 0 puntos	15
	Utiliza de manera adecuada Spark SQL para consultas estructuradas con tablas temporales.	Muestra dificultades para utilizar Spark SQL para consultas estructuradas con tablas temporales.	No utiliza Spark SQL para consultas estructuradas con tablas temporales.	
3. Desarrollo de modelos supervisados.	25 – 23 puntos	22 – 20 puntos	19 – 0 puntos	25
	Desarrolla modelos supervisados (regresión y clasificación) con Spark MLlib.	Presenta con detalles el desarrollo de modelos supervisados con Spark MLlib.	Desarrolla de manera incorrecta modelos supervisados.	
4. Aplicación de modelos no supervisados.	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	20
	Aplica de manera correcta modelos no supervisados (K-Means y evaluación con WSSSE, MAE, RMSE).	Aplica con algunos detalles los modelos no supervisados.	Aplica de manera incorrecta los modelos no supervisados.	
5. Construcción y análisis de grafos.	15 – 13 puntos	12 – 10 puntos	9 – 0 puntos	15
	Construye y analiza de manera correcta grafos usando GraphX (relaciones y nodos influyentes).	Construye, pero no analiza de manera adecuada grafos usando GraphX (relaciones y nodos influyentes).	Realiza una construcción y análisis de grafos incorrecto.	
6. Documentación.	10 – 8 puntos	7 – 5 puntos	4 – 0 puntos	10
	Presenta una documentación clara, reflexiones críticas y da respuestas a preguntas planteadas.	Presenta una documentación clara, incluye reflexiones críticas, pero no contesta todas las preguntas planteadas.	No presenta una documentación clara, no incluye reflexiones críticas ni contesta todas las preguntas planteadas.	
<b>TOTAL</b>				<b>100%</b>

## Prácticas de bienestar

### Práctica 1

<b>Nombre de la práctica</b>	Un momento para respirar
<b>Descripción de la práctica</b>	Aprenderás a respirar por la nariz y a tranquilizar tu mente.
<b>Palabras clave</b>	Fortalezas de carácter, autorregulación.  La autorregulación, también percibida como control, es una fortaleza de carácter muy importante dentro de la psicología positiva. Este concepto implica regular lo que uno siente y hace, ser disciplinado, así como mantener un control sobre los apetitos y, en especial, sobre las emociones.  En la actualidad vivimos situaciones muy estresantes que provocan que nuestra reacción instintiva y natural ante ellas sea estallar en ira. Y las consecuencias de este comportamiento se quedan en nosotros, pero también pueden llegar a afectar a terceros.  A continuación, se presenta un ejercicio que te ayudará a cultivar la fortaleza de autorregulación:
<b>Instrucciones para el aprendizador</b>	<ol style="list-style-type: none"> <li>1. Toma dos minutos de tu tiempo, siéntate en un lugar cómodo donde no haya mucho ruido que te distraiga.</li> <li>2. Escucha música de relajación (crea tu propio ambiente de meditación).</li> <li>3. Comienza a respirar y exhalar por nariz.</li> <li>4. Trata de que tu respiración y exhalación duren el mismo tiempo.</li> <li>5. Fija tu mente en tu respiración, en cómo entra y sale el aire de tu cuerpo.</li> </ol> <p>Repite durante dos minutos.</p> <p>Te recomendamos que sí, durante este periodo, algún pensamiento (olvidé algo en la oficina, más tarde tengo que hacer tal actividad, etc.) llega a tu mente, solo déjalo pasar y regresa a la concentración en tu respiración.</p> <p>Al finalizar los dos minutos sentirás paz en tu ser. Comienza a hacer este ejercicio de respiración y meditación todos los días y poco a poco ve aumentando el tiempo.</p>
<b>Fuente</b>	Conferencia Rosalinda Ballesteros.

## Práctica 2

<b>Nombre de la práctica</b>	Fomentando la atención plena
<b>Descripción de la práctica</b>	Realizarás ejercicios de meditación para fomentar la atención plena en tus actividades diarias.
<b>Palabras clave</b>	Atención plena, fortalezas de carácter, autorregulación.
<b>Instrucciones para el aprendizador</b>	<p>La meditación es una herramienta que ayuda a mejorar el desempeño de cualquier persona, ya que fomenta el desarrollo de la atención plena en una sola actividad. Para fomentar la atención plena y lograr cada vez más estar en una zona de concentración mientras realizas tus actividades cotidianas, lleva a cabo los siguientes ejercicios de meditación:</p> <p>Encuentra en algún momento del día cinco minutos para ti, siéntate en un lugar cómodo, donde no tengas distracciones.</p> <ol style="list-style-type: none"> <li>1. Haz tres respiraciones profundas por la nariz y exhala por la nariz.</li> <li>2. Comienza a hacer un repaso de tu día, de lo que más te acuerdes, por ejemplo, te levantaste, ¿qué hiciste?, ¿desayunaste?, ¿te bañaste?, ¿diste los buenos días?, etcétera. Si desayunaste, ¿qué fue lo que desayunaste?, ¿te gustó?, ¿tomaste tu alimento despacio o apurado? Si estabas apurado, ¿qué era lo que te tenía en esa situación?</li> <li>3. Sigue meditando en lo que te acuerdes: ¿te molestase con alguien?, ¿por qué?, ¿qué fue lo que pasó?, ¿crees que era posible haber reaccionado de alguna manera más pacífica?</li> </ol> <p>Con este ejercicio te darás cuenta de que reaccionamos o hacemos cosas de manera automática. Algunas veces si estamos más conscientes y presentes, podemos tener otra actitud sin que alguna situación nos afecte demasiado.</p>
<b>Fuente</b>	Eby, D. (s.f.). <i>Creativity and Flow Psychology</i> . Recuperado de <a href="http://talentdevelop.com/articles/Page8.html">http://talentdevelop.com/articles/Page8.html</a>

**Práctica 03**

<b>Nombre de la práctica</b>	Experiencias difíciles
<b>Descripción de la práctica</b>	Analizarás las estrategias que seguiste para afrontar problemáticas y cómo aprendiste de esos sucesos.
<b>Palabras clave</b>	Resiliencia
<b>Instrucciones para el aprendizador</b>	<p>Todos hemos pasado por situaciones complejas, tanto en el ámbito laboral, como en el familiar y personal. La manera en que enfrentamos dichos obstáculos es muy diferente, algunas personas continúan con su vida sin problema alguno, a otras tantas se les complica esa transición, también hay quienes no pueden sobreponerse a las experiencias difíciles.</p> <p>La resiliencia es la capacidad de reponerse tras la adversidad, de recuperarse después de vivir experiencias difíciles, dolorosas o traumáticas. Para algunos la resiliencia implica salir adelante después de una situación muy dura, y además crecer o ser mejor a raíz de esta experiencia. (Tarragona, 2012)</p> <p>La siguiente práctica te ayudará a fomentar esta importante cualidad:</p> <ol style="list-style-type: none"> <li>1. Crea una tabla con tres columnas y cinco filas.</li> <li>2. En la primera columna escribe un evento difícil o desagradable al que te hayas enfrentado en tu vida.</li> <li>3. En la segunda columna menciona cuáles son tus creencias sobre esa adversidad.</li> <li>4. En la tercera columna describe las consecuencias que tiene esa creencia.</li> <li>5. Cuando termines, lee toda la tabla y reflexiona sobre cómo te ha cambiado cada evento y cómo lo enfrentaste.</li> <li>6. Al final escribe cómo enfrentarías cada evento en la actualidad.</li> </ol>
<b>Fuente</b>	<ul style="list-style-type: none"> <li>• Metodología ABC.</li> <li>• Fundamentos de psicología positiva.</li> </ul>

**Práctica 04**

<b>Nombre de la práctica</b>	Concentrarse en lo positivo
<b>Descripción de la práctica</b>	Analizarás sucesos que te hayan ocurrido recientemente, buscando orientar el análisis hacia las consecuencias positivas.
<b>Palabras clave</b>	Resiliencia y esperanza.
<b>Instrucciones para el aprendizador</b>	<p>¿Qué es lo primero que piensas cuando recibes una noticia inesperada?, o bien, ¿qué te imaginas cuando un acontecimiento complejo se presenta ante ti?</p> <p>La mayoría de las personas se concentra en el peor de los escenarios independientemente del tipo de noticia que reciban. Martin Seligman sugiere hacer un breve ejercicio para fomentar la resiliencia y la esperanza con base en la premisa antes señalada:</p> <ol style="list-style-type: none"> <li>1. Piensa en una noticia reciente que hayas recibido y que creas que es negativa para ti.</li> <li>2. Luego de analizarla, haz una tabla con tres columnas. En la primera, señala cuál sería el peor de los escenarios posibles que pudiera resultar de esa noticia; en la segunda columna señala cuál sería el mejor de los escenarios posibles, y en la última, cuál es el escenario que realmente tiene mayor probabilidad de ocurrir.</li> <li>3. Reflexiona sobre los tres escenarios, ¿cómo enfrentarías cada uno?</li> </ol> <p>Procura repetir este ejercicio cada vez que sientas que te enfrentas a una situación complicada. Hacerlo te dará perspectiva y te ayudará a cultivar tu resiliencia.</p>
<b>Fuente</b>	Seligman, M. (2011). <i>Building Resilience</i> . Recuperado de <a href="https://hbr.org/2011/04/building-resilience">https://hbr.org/2011/04/building-resilience</a>

**Práctica 05**

<b>Nombre de la práctica</b>	Crecimiento postraumático
<b>Descripción de la práctica</b>	Harás un recuento de las situaciones difíciles a las que te has enfrentado y reflexionarás sobre lo positivo que surgió de ellas.
<b>Palabras clave</b>	Resiliencia
	<p>La resiliencia es la capacidad de reponerse tras la adversidad, de recuperarse después de vivir experiencias difíciles, dolorosas o traumáticas. Para algunos la resiliencia implica salir adelante después de una situación muy dura, y además crecer o ser mejor a raíz de esta experiencia. (Tarragona, 2012)</p>
	<p>La siguiente práctica te ayudará a fomentar esta importante cualidad:</p> <ol style="list-style-type: none"> <li>1. Escribe acerca de un momento en el que enfrentaste una adversidad significativa o pérdida.</li> <li>2. Primero escribe acerca de las puertas que se te cerraron debido a esa adversidad o pérdida, ¿qué perdiste?</li> <li>3. Después, escribe acerca de las puertas que se abrieron al termino o como secuela de esa adversidad o pérdida.</li> <li>4. ¿Hay nuevas maneras de actuar, pensar o relacionarse que son más probables que sucedan ahora?</li> </ol>
<b>Instrucciones para el aprendizador</b>	
<b>Fuente</b>	Ejercicio contribuido por Taylor Kreiss de University of Pennsylvania Positive Psychology Center, y basado en el libro: <i>A Primer in Positive Psychology</i> de Christopher Peterson.

**Práctica 06**

<b>Nombre de la práctica</b>	La mejor versión de ti mismo
<b>Descripción de la práctica</b>	Escribirás acerca de la mejor versión posible de ti mismo durante al menos 20 minutos.
<b>Palabras clave</b>	Emociones positivas, fortalezas de carácter, autorregulación y esperanza.
<b>Instrucciones para el aprendizador</b>	<p>Imagina que han pasado 20 años y has crecido en todas las áreas o maneras que te gustaría crecer y las cosas te han salido tan bien como te las imaginaste.</p> <ul style="list-style-type: none"> <li>• ¿Cómo es esa mejor versión de ti mismo?</li> <li>• ¿Qué hace él o ella cotidianamente?</li> <li>• ¿Qué dicen los demás acerca de él o ella?</li> </ul> <p>No es necesario que compartas este escrito, ya que el objetivo de esta reflexión es enfocarse en la experiencia que viviste mientras reflexionabas en esa mejor versión posible de ti mismo.</p>
<b>Fuente</b>	Ejercicio contribuido por Taylor Kreiss de University of Pennsylvania Positive Psychology Center, y basado en el libro <i>A Primer in Positive Psychology</i> de Christopher Peterson.

**Práctica 07**

<b>Nombre de la práctica</b>	Obtener lo que quieres
<b>Descripción de la práctica</b>	Reflexionarás sobre alguna meta que desees alcanzar y propondrás una forma de conseguirla.
<b>Palabras clave</b>	Logro, involucramiento, fortalezas de carácter, esperanza, autorregulación, metas y objetivos a largo plazo.
<b>Instrucciones para el aprendizador</b>	<p>Es importante tener una idea clara de lo que desees lograr a corto, mediano y largo plazo, pues te ayuda a seguir un camino trazado.</p> <p>Para que puedas generar esta guía, responde las siguientes preguntas:</p> <ol style="list-style-type: none"> <li>1. ¿Qué quieres lograr? Al trazar tu meta, procura que esta sea específica, medible, alineada, realista, retadora y con una fecha para lograrla. Piensa en algo y utiliza el método SMART para definirla.</li> <li>2. ¿Qué te impide que lo tengas en este momento?</li> <li>3. ¿Qué sufrimiento estás experimentando en tu vida por no tenerlo en este momento?</li> <li>4. ¿Qué placer, involucramiento, relación, significado o logro tendrías en tu vida si tuvieras eso en este momento?</li> <li>5. ¿Qué hábitos te detienen o no te dejan avanzar hacia eso que quieres?</li> <li>6. ¿Qué nuevos hábitos podrías generar para ayudarte a obtener lo que quieres?</li> <li>7. Menciona dos cosas que podrías hacer para romper con los hábitos que no te permiten avanzar hacia lo que quieres y generar hábitos nuevos.</li> <li>8. ¿Te comprometes a hacer esas dos cosas? Si es así, ¿cuándo las harás?</li> </ol> <p>Escribe tus respuestas en un sitio donde puedas verlas siempre.</p>
<b>Fuente</b>	Ejercicio contribuido por Taylor Kreiss de University of Pennsylvania Positive Psychology Center, y basado en el libro <i>A Primer in Positive Psychology</i> de Christopher Peterson.

**Práctica 08**

<b>Nombre de la práctica</b>	Felicidad en el trabajo
<b>Descripción de la práctica</b>	Reflexionarás sobre las distintas dimensiones de tu vida cotidiana, enfocando el análisis a cómo fomentar un estado de ánimo y relaciones positivas en el ámbito laboral.
<b>Palabras clave</b>	Involucramiento, emociones positivas, relaciones positivas.
<b>Instrucciones para el aprendizador</b>	<p>Elegir con consciencia maneras de incrementar la felicidad en el trabajo hace la diferencia en cómo nos sentimos y qué tan bien nos desempeñamos. En lugar de quejarnos del trabajo, ¿por qué no pensar en cómo podemos obtener mayor felicidad de lo que hacemos?</p> <p>Estar más involucrados en lo que hacemos contribuye a nuestra felicidad y bienestar, y nos lleva a un mejor desempeño y productividad. A manera de reflexión, responde las siguientes preguntas enfocadas en distintas dimensiones de tu vida:</p> <ul style="list-style-type: none"> <li>• <b>Dar:</b> ¿cómo estoy apoyando a mis colaboradores, compañeros, líderes, proveedores y clientes?</li> <li>• <b>Relaciones:</b> ¿cómo puedo mejorar mis relaciones en el trabajo?, ¿cómo logro un balance entre la vida laboral y familiar?</li> <li>• <b>Ejercicio:</b> ¿cómo puedo integrar la actividad física dentro de mis actividades diarias?, ¿cómo aseguro que estoy comiendo bien y descansando lo suficiente?</li> <li>• <b>Conciencia:</b> ¿cómo puedo construir momentos de atención plena en mi día laboral?</li> <li>• <b>Ensayo:</b> ¿qué habilidades estoy construyendo?, ¿qué cosas nuevas he experimentado?</li> <li>• <b>Dirección:</b> ¿cuáles son mis metas laborales hoy, esta semana y este año?, ¿cómo caben y contribuyen mis metas laborales con mis metas de vida?, ¿me ayudan a desarrollar mis competencias en la construcción de mis relaciones?, ¿cómo contribuyo con lo anterior a ayudar a otros?, ¿cómo se pueden alinear mis metas laborales con las de mi equipo y la organización?</li> <li>• <b>Resiliencia:</b> ¿cuáles son mis tácticas para lidiar con los retos difíciles en el trabajo?, ¿me estoy enfocando en lo que puedo controlar?, ¿necesito pedir ayuda a otros?, ¿hay alguien a mi alrededor que requiere de mi ayuda?</li> <li>• <b>Emoción:</b> ¿qué cosas, aunque sean pequeñas, puedo encontrar que me pueden hacer sentir bien en mi trabajo hoy?, ¿qué me ha hecho sonreír?</li> </ul>
<b>Fuente</b>	Tomado del Catálogo de actividades para profesores.

**Práctica 9**

<b>Nombre de la práctica</b>	Interacciones positivas
<b>Descripción de la práctica</b>	Reflexionarás sobre las cualidades positivas que aprecias de las personas con las que interactúas todos los días.
<b>Palabras clave</b>	<p>Relaciones positivas</p> <p>Puedes obtener mayor gozo de los momentos que compartes con tus colegas si te tomas el tiempo para pensar en lo que valoras y aprecias de ellos. Diversas investigaciones muestran que enfocarse en lo positivo que sucede cada día ayuda a incrementar la felicidad y lo mismo aplica a todas nuestras relaciones cercanas.</p> <p>El psicólogo John Gottman sugiere que, para tener relaciones felices con alguna persona, es necesario aspirar a tener cinco interacciones positivas por cada interacción negativa que se tenga con ella. Enfócate en tus compañeros o colegas y piensa en las siguientes preguntas. En cada caso, anota ejemplos específicos.</p> <ol style="list-style-type: none"> <li>1. ¿Qué te atrajo de tus compañeros cuando se conocieron?</li> <li>2. ¿Qué cosas han disfrutado al hacerlas juntos?</li> <li>3. ¿Qué cosas realmente aprecias de ellos en este momento?</li> <li>4. ¿Cuáles son sus fortalezas?</li> </ol>
<b>Instrucciones para el aprendizador</b>	<p>Ahora, lo más importante es que cuando estés con tus compañeros te tomes el tiempo para darte cuenta y reconocer estas cualidades, sus fortalezas y las cosas que ellos hacen que realmente aprecies, así como los momentos agradables que han compartido.</p> <p>Piensa en estas declaraciones:</p> <ul style="list-style-type: none"> <li>• “Realmente me encanta cuando ellos...”.</li> <li>• “Son tan buenos para...”.</li> <li>• “Viéndolos hacer..., me recuerda ese fantástico día cuando nosotros...”.</li> </ul> <p>Aunque realizar dicho análisis con todas las personas que conoces resulta poco práctico, puedes usar los mismos principios para mejorar tus relaciones en general. Por ejemplo, antes de pasar tiempo con alguien tómate un momento para pensar en aquellas cosas que te gustan, aprecias o admiras de esa persona o cómo te hacen sentir bien. Asimismo, después de pasar tiempo con esa persona, piensa en las cosas que apreciaste o lo que disfrutaste del tiempo que pasaron juntos.</p>
<b>Fuente</b>	Basado en el Catálogo de actividades para profesores.

**Práctica 10**

<b>Nombre de la práctica</b>	Las fortalezas se muestran en nuestras historias
<b>Descripción de la práctica</b>	Reflexionarás sobre las fortalezas de carácter que aplicaste en una situación.
<b>Palabras clave</b>	Fortalezas de carácter
<b>Instrucciones para el aprendizador</b>	<p>Antes de comenzar el ejercicio, ¿sabes cuáles son las fortalezas de carácter?          Consulta la descripción de las 24 fortalezas de carácter en la siguiente liga:</p> <p style="text-align: center;"><a href="http://www.viacharacter.org/www/Character-Strengths/VIA-Classification">http://www.viacharacter.org/www/Character-Strengths/VIA-Classification</a></p> <p style="text-align: center;">El enlace anterior es externo a la Universidad Tecmilenio, al acceder a este considera que debes apegarte a sus términos y condiciones.</p> <p>Luego de que leas cuáles son las fortalezas de carácter, realiza lo que se pide a continuación:</p> <ol style="list-style-type: none"> <li>1. Describe detalladamente en un texto, una anécdota en la que hayas llevado a cabo alguna acción de la mejor manera posible, o bien, que hayas actuado por encima de lo ordinario. Procura enfocarlo al entorno laboral.</li> <li>2. Puede ser cualquier suceso que te haya marcado por la manera en que te desarrollaste.</li> <li>3. Señala en tu descripción: ¿qué ocurrió?, ¿qué papel jugaste en el suceso?, ¿qué acciones llevaste a cabo que fueron de utilidad para ti y para los demás?</li> <li>4. Luego de que hayas terminado de escribir, lee tu texto y subraya las palabras y oraciones que te den una idea sobre cómo usaste cualquiera de las 24 fortalezas de carácter.</li> <li>5. Observa y clasifica cuáles son las fortalezas que usaste en tu anécdota. Reflexiona sobre el impacto que pueden tener en tu desempeño cotidiano.</li> </ol>
<b>Fuente</b>	Niemiec, R. (2016). <i>How to Assess Your Strengths: 5 Tactics for Self-Growth</i> . Recuperado de <a href="https://www.psychologytoday.com/us/blog/what-matters-most/201603/how-assess-your-strengths-5-tactics-self-growth">https://www.psychologytoday.com/us/blog/what-matters-most/201603/how-assess-your-strengths-5-tactics-self-growth</a>

**Práctica 11**

<b>Nombre de la práctica</b>	Tus fortalezas en los ojos del otro
<b>Descripción de la práctica</b>	Reflexionarás sobre la percepción que otros tienen sobre tus fortalezas de carácter.
<b>Palabras clave</b>	Fortalezas de carácter
<b>Instrucciones para el aprendizador</b>	<p>¿Recuerdas alguna ocasión en la que hablaste con algún colega y este te reveló algo positivo que piensa de ti? Cuando esto ocurre, deja huella en nuestros comportamientos y acciones, pues nos damos cuenta de que las personas tienen percepciones sobre nuestras fortalezas que nosotros mismos no percibimos. Haz lo siguiente:</p> <ol style="list-style-type: none"> <li>1. Piensa sobre alguna vez que algún compañero de trabajo te compartió lo que piensa de ti y que te haya sorprendido.</li> <li>2. Piensa en lo siguiente: ¿qué fue lo que te llamó más la atención?, ¿qué fortalezas vio en ti que pensaste que no tenías tan desarrolladas?</li> <li>3. Por último, señala en un texto por qué consideras que esta revelación te causó tanto impacto, así como la manera en que te ayudó a cultivar tus fortalezas de carácter.</li> </ol>
<b>Fuente</b>	<p>Niemiec, R. (2016). <i>How to Assess Your Strengths: 5 Tactics for Self-Growth</i>. Recuperado de <a href="https://www.psychologytoday.com/us/blog/what-matters-most/201603/how-assess-your-strengths-5-tactics-self-growth">https://www.psychologytoday.com/us/blog/what-matters-most/201603/how-assess-your-strengths-5-tactics-self-growth</a></p>

## Práctica 12

<b>Nombre de la práctica</b>	Plantea tus objetivos como metas de aproximación y replantea tus metas de evitación
<b>Descripción de la práctica</b>	Establecerás diferentes tipos de metas y encontrarás la mejor manera de conseguirlas.
<b>Palabras clave</b>	<p>Objetivos, metas y planes.</p> <p>La autora Bridget Grenville-Cleave (2012) comenta que hay dos tipos de metas:</p> <ol style="list-style-type: none"> <li>1. <b>Metas de aproximación (<i>approach</i>):</b> son las metas con resultados positivos (deseables, placenteros, benéficos o que nos gustaría tener) y hacia las cuales trabajamos.</li> <li>2. <b>Metas de evitación (<i>avoidance</i>):</b> son las metas con resultados negativos (indeseables, dolorosos, dañinos, o nos disgustan) y en las cuales trabajamos para evitarlas.</li> </ol>
<b>Instrucciones para el aprendizador</b>	<p><b>Ejemplo:</b></p> <p>Meta de aproximación:</p> <ul style="list-style-type: none"> <li>• Ser más eficiente.</li> <li>• Ser amigable y extrovertido en reuniones.</li> <li>• Asumir el rol de líder en el trabajo.</li> </ul> <p>Meta de evitación:</p> <ul style="list-style-type: none"> <li>• Dejar de aplazar.</li> <li>• Dejar de ser tan tímido en las reuniones.</li> <li>• No pasar desapercibido en el trabajo.</li> </ul> <p>Las investigaciones que se han realizado respecto a estos tipos de metas muestran que perseguir metas de evitación resulta en un detrimento del bienestar. Estos descubrimientos sugieren que el establecer metas de aproximación o replantear las metas de evitación es benéfico.</p> <p>Reflexiona lo siguiente:</p> <ul style="list-style-type: none"> <li>• ¿Qué tipo de metas te has planteado tú?</li> <li>• ¿Hay algunas metas que puedas replantear en una forma más positiva?</li> <li>• ¿Cuándo las tendrás listas?</li> </ul>
<b>Fuente</b>	Grenville, B. (2012). <i>GOAL-SETTING SECRETS</i> . Recuperado de <a href="http://positivepsychologynews.com/news/bridget-grenville-cleave/2012013120696">http://positivepsychologynews.com/news/bridget-grenville-cleave/2012013120696</a>