



Desarrollo full stack

Guía para el profesor
Clave LSTI2313



Contenido

Datos generales.....	3
Competencia global.....	3
Competencias esenciales.....	3
Introducción.....	4
Información general.....	5
Calendario de entregas.....	8
Temario.....	9
Preguntas más frecuentes.....	11
Recomendaciones para la explicación de los temas, actividades y reto.....	12
Rúbrica del avance del reto (fase I).....	29
Rúbrica de la entrega final del reto (fase II).....	31
Prácticas de bienestar.....	33

Datos generales

Nombre del certificado: Desarrollo full stack

Nivel: Profesional

Modalidad: Presencial

Clave: LSTI2313

Competencia global

Desarrolla aplicaciones web completas desde la planificación y diseño hasta la implementación y mantenimiento, utilizando lenguajes de programación y *frameworks* modernos, garantizando soluciones eficientes, escalables y seguras que satisfagan las necesidades del cliente y usuario final en un entorno colaborativo y dinámico.

Competencias esenciales

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

Introducción

El certificado en Desarrollo full stack te capacitará para diseñar y desarrollar aplicaciones web completas, abarcando las etapas de planificación, diseño, implementación y mantenimiento. Este certificado integra conocimientos sobre *frontend*, *backend*, bases de datos, diseño de interfaces y despliegue en la nube, garantizando la creación de soluciones eficientes, escalables y seguras que satisfagan tanto las necesidades del cliente como las expectativas del usuario final, dentro de un entorno colaborativo y dinámico.

Además, dominarás prácticas esenciales como la autenticación segura, la creación e integración de APIs RESTful y GraphQL, el despliegue en la nube y el diseño de interfaces centradas en el usuario. Estas habilidades no solo fortalecen tus capacidades técnicas, sino que también te posicionan como un desarrollador integral, capaz de gestionar proyectos de principio a fin y de aportar valor tangible en equipos multidisciplinarios y entornos colaborativos.

Información general

Metodología

El modelo académico MAPS se distingue por su carácter modular, apilable y personalizable, con un enfoque flexible y centrado en el estudiante. Implementamos técnicas didácticas que fomentan tanto la adquisición de conocimientos teóricos como su aplicación práctica, además del desarrollo de competencias profesionales altamente valoradas por los empleadores. A continuación, se presentan las técnicas didácticas y las características principales de nuestro modelo académico.

Técnicas didácticas

Aprendizaje basado en retos. El estudiante demuestra la adquisición de los conocimientos y los aplica mediante retos planteados.

Aprendizaje basado en proyectos. El estudiante demuestra la adquisición de los conocimientos y los aplica en la práctica, a través de proyectos que generen un impacto positivo en las organizaciones.

Aula invertida. Esta metodología fomenta el autoestudio fuera del entorno de clases, de manera que, cuando los estudiantes acceden al aula virtual, se impulse la interacción, la construcción conjunta del conocimiento, la generación de ideas y el desarrollo de competencias, con el apoyo de docentes expertos.

El **aprendizaje basado en retos** se implementa desde el primer hasta el quinto semestre, el **aprendizaje basado en proyectos** se aplica a partir del sexto semestre, y la metodología de **aula invertida** está presente en todos los certificados.

En las Semanas de Desarrollo Integral (SeDI) y en los certificados de idioma, únicamente se implementa la metodología de aula invertida.

Características

1. Certificados
 - a. El modelo se compone de certificados de especialidad, diseñados para fomentar el desarrollo y la adquisición de competencias demandadas por los principales empleadores del país a través del aprendizaje activo.
 - b. Todos los certificados se desarrollan en alianza con empresas de prestigio nacional e internacional, y/o con expertos que poseen conocimiento técnico y académico actual, requerido en diversas industrias, lo que garantiza el desarrollo de competencias profesionales.
 - c. En cada periodo, el estudiante cursa un máximo de dos certificados simultáneamente, lo que le brinda la oportunidad de profundizar en cada tema. Esto resulta particularmente valioso en cursos que exigen una comprensión detallada de teorías complejas, aplicaciones prácticas y habilidades analíticas avanzadas.

2. Duración

La duración de la licenciatura varía según el formato elegido. Los programas ejecutivos se completan en 15 bimestres y los programas semestrales en 8 semestres. Ambos formatos están conformados por los mismos certificados en sus respectivos mapas curriculares, lo que permite a los estudiantes transitar entre ambas modalidades según sus necesidades.

3. Flexibilidad

Este modelo fomenta la participación de los estudiantes, permitiéndoles personalizar su experiencia de aprendizaje según sus intereses y necesidades individuales. Esta personalización no solo facilita un mayor compromiso y motivación, sino que también prepara a los estudiantes para enfrentar retos específicos de su futuro campo profesional, lo que incrementa su empleabilidad y éxito académico.

4. Credenciales apilables

La idea detrás de estas credenciales es proporcionar un esquema de capacitación y aprendizaje para los estudiantes, de manera que puedan avanzar rápidamente en el proceso educativo, adquiriendo habilidades aplicables al entorno laboral. Por lo tanto, las credenciales pueden ser apiladas para cumplir con el estándar de un programa de grado tradicional.

5. Insignias digitales

Las insignias digitales permiten registrar la formación académica de los estudiantes y sus logros. Una de las ventajas de las insignias digitales es que, mediante los metadatos, se pueden obtener detalles de las competencias adquiridas, la institución que otorga la insignia, y un reconocimiento visual que puede compartirse en redes sociales o profesionales.

6. Diferenciadores del modelo

- a. **Certificados de lengua extranjera:** se ofrecen certificados para adquirir o reforzar el dominio de una lengua extranjera, así como certificados impartidos en una lengua extranjera y específicos de la disciplina, con el objetivo de atender las demandas de los empleadores.
- b. **Semanas de Desarrollo Integral:** unidades de aprendizaje transversal, diseñadas para ofrecer una experiencia inmersiva que desarrolla competencias humanas, profesionales y de bienestar.
- c. **Periodos de Skilling:** un periodo complementario en el que el estudiante puede realizar actividades que complementan su formación académica. Estas actividades son opcionales y personalizadas, ya que el estudiante las elige según sus intereses profesionales y personales.
- d. **Estancia empresarial al final del programa de estudios:** los estudiantes podrán elegir entre tres opciones de estancia empresarial, que incluyen: gestión de proyectos, emprendimiento y desarrollo sostenible.

Bibliografía y software

Bibliografía opcional

- Bouchebra, A. (2022). *Full Stack Development with Angular and GraphQL: Learn to Build Scalable Monorepo and a Complete Angular App Using Apollo, Lerna, and GraphQL*. Reino Unido: Packt Publishing. ISBN: 978-1800202467
- Derks, R. (2019). *React Projects: Build 12 Real-world Applications From Scratch Using React, React Native, and React 360*. Reino Unido: Packt Publishing. ISBN: 978-1789954937

Software:

- Angular. (s.f.). *Build your first Angular app*. Recuperado de <https://angular.dev/tutorials/first-app>
- Docker Compose. (s.f.). *Docker Compose*. Recuperado de <https://docs.docker.com/compose/>
- Git. (s.f.). *git fast-version-control*. Recuperado de <https://git-scm.com/>

- MySQL. (s.f.). *Downloads*. Recuperado de <https://www.mysql.com/>
- MongoDB (s.f.). *MongoDB Atlas*. Recuperado de https://www.mongodb.com/lp/cloud/atlas/try3?utm_content=rlsapostreg&utm_source=google&utm_campaign=search_gs_pl_evergreen_atlas_general_retarget-brand-postreg_gic-null_amers-all_ps-all_desktop_eng_lead&utm_term=&utm_medium=cpc_paid_search&utm_ad=&utm_ad_campaign_id=14412646452&adgroup=131761126052&cq_cmp=14412646452&gad_source=1&gclid=CjwKCAiA3Na5BhAZEiwAzrfagCsmxXN8gX-NAMU1RFHBDkiA1ADYZFPDHTIINZkbaWupAeqKzoFpmBoCA6AQAvD_BwE
- Node.js. (s.f.). *Download Node.js*. Recuperado de <https://nodejs.org/en/download/package-manager/current>
- React. (s.f.). *Installation*. Recuperado de <https://react.dev/learn/installation>
- Vue.js. (s.f.). *The Progressive JavaScript Framework*. Recuperado de <https://vuejs.org/>
- Visual Studio Code. (s.f.). *Documentación de Visual Studio Code*. Recuperado de <https://code.visualstudio.com/docs/?dv=win64user>

Evaluación

La evaluación combina los siguientes elementos:

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

A continuación, se presenta el detalle de la evaluación:

Semana	Evaluable	Ponderación
1	Actividad I	6%
2	Actividad II	6%
3	Avance del reto	25%
4	Actividad III	6%
5	Actividad IV	6%
6	Certificación	6%
7	Entrega final del reto	35%
8 Semana de Assesment	Presentación del reto	10%
Total		100%

Actividades y fases del reto

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

Con el fin de fomentar el dinamismo y la interacción entre los participantes en diversos formatos, el profesor alternará, durante las sesiones, intervenciones individuales, plenarias y grupales. Estas actividades enriquecerán tus perspectivas 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 reto 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. Lo anterior con la intención de que desde 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 reto o el contenido, puedes contactar a tu docente a través de los medios que se te indiquen.

Calendario de entregas semestral

Semana	Evaluable
1	Actividad I
2	Actividad II
3	Avance del reto
4	Actividad III
5	Actividad IV
6	Certificación
7	Entrega final del reto
8 Semana de Assesment	Presentación del reto

Temario

Tema 1. Introducción al desarrollo full stack

- 1.1 Conceptos básicos de desarrollo web
- 1.2 Roles del desarrollador full stack
- 1.3 Herramientas y entornos de desarrollo

Tema 2. HTML y CSS

- 2.1 Fundamentos de HTML
- 2.2 Uso de CSS

Tema 3. JavaScript básico

- 3.1 Fundamentos de JavaScript
- 3.2 Manipulación del DOM

Tema 4. JavaScript avanzado

- 4.1 ES6+ y características modernas
- 4.2 Programación orientada a objetos (OOP)

Tema 5. Frameworks de frontend

- 5.1 Introducción a React.js
- 5.2 Generalidades y ejemplos de Angular
- 5.3 Generalidades y ejemplos de Vue.js

Tema 6. Diseño y experiencia de usuario (UI/UX)

- 6.1 Diseño de interfaz de usuario (UI)
- 6.2 Experiencia de usuario (UX)

Tema 7. Desarrollo backend con Node.js

- 7.1 Event Loop y asincronía
- 7.2 Creación de un servidor básico
- 7.3 Módulos nativos de Node.js (fs y http)
- 7.4 Manejadores de errores y *debugging*

Tema 8. Express.js

- 8.1 Rutas y *middleware*
- 8.2 Manejo de peticiones y respuestas
- 8.3 Autenticación y sesiones
- 8.4 *Middleware* personalizado

Tema 9. Bases de datos (operaciones CRUD)

- 9.1 Bases de datos SQL
- 9.2 Bases de datos NoSQL

Tema 10. Autenticación y autorización

- 10.1 Autenticación con JWT
- 10.2 Utilización de OAuth para autenticación

Tema 11. APIs y servicios web

- 11.1 Desarrollando con RESTful APIs
- 11.2 Implementando GraphQL en proyectos full stack

Tema 12. Infraestructura en la nube para full stack

- 12.1 Despliegue en la nube
- 12.2 Contenedores y orquestación

Tema 13. Pruebas y depuración

- 13.1 Casos de prueba en full stack
- 13.2 Depuración de código

Tema 14. Desarrollo móvil y aplicaciones en tiempo real

- 14.1 Configuración del entorno de desarrollo
- 14.2 Componentes y navegación con React Native
- 14.3 Integración de APIs nativas
- 14.4 Despliegue en iOS y Android

Preguntas más 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 de la cantidad de sesiones y el tiempo de cada sesión en las semanas?

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 de 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 reto.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Comienza con una breve contextualización de la disciplina dentro de la ingeniería de *software*, resaltando la importancia de dominar tanto el frontend como el backend. Esto permite a los aprendedores visualizar de forma integral el papel del desarrollador full stack en proyectos reales.
- Explica los conceptos de frontend, backend, bases de datos, control de versiones, seguridad y despliegue, utiliza ejemplos prácticos como el desarrollo de una tienda en línea. Un enfoque visual es esencial: utiliza diagramas o esquemas que muestren cómo interactúan estas tecnologías; esto facilitará la comprensión inicial de conceptos abstractos.
- Es probable que surjan dudas en la diferencia de roles y herramientas específicas del frontend y backend, así como en la integración entre ambos. Para aclarar esto, es recomendable realizar una demostración en vivo de una aplicación simple, donde se muestre cómo el frontend envía datos al backend y cómo este los procesa y almacena en una base de datos.
- Aborda los conceptos relacionados con el control de versiones y el despliegue continuo con ejemplos prácticos, como la creación de ramas en Git y su posterior integración.

Recomendaciones: durante la sesión, se sugiere fomentar la participación a través de preguntas abiertas como: ¿qué aspectos consideran más desafiantes al integrar frontend y backend?" para conectar los conceptos con las experiencias previas de los aprendedores.

Realiza un resumen comparativo de los *stacks* más populares (MERN, MEAN) para que los aprendedores comprendan cómo elegir las tecnologías adecuadas según las necesidades del proyecto. Como *tip*, integra un breve ejercicio práctico donde los aprendedores configuren un entorno básico de frontend y backend, con el objetivo de reforzar la conexión entre la teoría y la práctica.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Aborda el tema "HTML y CSS" con un enfoque práctico, enfatizando la relación entre la teoría y la aplicación en proyectos reales.
- Explica el rol de ambos lenguajes en el desarrollo full stack, destacando que HTML proporciona la estructura base de las páginas web y CSS define su apariencia. Utiliza ejemplos sencillos para mostrar cómo se conectan ambos lenguajes, como la construcción de una página básica que evoluciona desde su estructura HTML hasta su estilización con CSS.
- Realiza actividades donde los aprendedores repliquen componentes web comunes, como encabezados, formularios y botones, aplicando buenas prácticas como el uso de atributos semánticos y estilos adaptativos.
- El área que podría generar más dudas es la combinación de selectores CSS con propiedades específicas y cómo estos interactúan con el DOM. Por ejemplo, la diferencia

entre selectores de clase e ID, o cómo usar pseudoclasas como `:hover` para efectos interactivos. Estas dudas se pueden aclarar haciendo ejercicios guiados donde los aprendedores modifiquen estilos en tiempo real y observen los resultados inmediatamente.

- Explica de forma clara y detallada la semántica de HTML, la aplicación de CSS externo versus interno y el diseño responsivo mediante *media queries*.

Recomendaciones: para facilitar el aprendizaje, se debe promover la exploración visual del código mediante navegadores web y herramientas como el inspector de elementos. Esto permitirá a los aprendedores visualizar directamente cómo cada etiqueta o regla CSS afecta el diseño.

Utiliza analogías visuales (ejemplo, "HTML es el esqueleto y CSS es la piel") y herramientas como CodePen para demostrar ejemplos en vivo. Además, promover el uso de estándares de accesibilidad y optimización del código desde el inicio, esto fomentará una comprensión sólida y profesional del tema.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Explica los conceptos fundamentales con ejemplos prácticos y aplicables al desarrollo web. Es crucial iniciar con una introducción clara sobre la relevancia de JavaScript en el desarrollo full stack, conectando sus capacidades frontend y backend para crear aplicaciones modernas.
- La clase debe ser dinámica, utilizando ejercicios en tiempo real que permitan a los aprendedores experimentar con la sintaxis del lenguaje y sus principales características, como variables, tipos de datos, operadores, y estructuras de control.
- Divide el tema en módulos progresivos, comenzando con los fundamentos y avanzando hacia conceptos como funciones, bucles y manejo de eventos. Además, usa analogías prácticas y casos reales, como la creación de elementos dinámicos en una página web, para mantener el interés y relacionar los temas con problemas cotidianos del desarrollo web.
- Es probable que las dudas puedan surgir en los conceptos como el alcance de las variables (*var*, *let* y *const*), la asincronía con promesas y *async/await*, y la manipulación del DOM. Para aclarar estas dudas, es útil implementar ejercicios guiados que incluyan errores deliberados para que los aprendedores los detecten y comprendan sus soluciones.
- Es clave emplear herramientas visuales, como diagramas para explicar el flujo de ejecución de los bucles o las estructuras condicionales, además de proporcionar resúmenes de los conceptos al final de cada subtema.
- Explica de forma clara y detallada la diferencia entre `==` y `===`, el uso de *addEventListener* para la interacción con eventos, y las funciones flecha, destacando su relación con el contexto *this*.

Recomendaciones: se recomienda usar entornos interactivos como CodePen o JSFiddle durante la clase, para que los aprendedores puedan ver inmediatamente el impacto de sus cambios en el código. Además, incluir un pequeño proyecto integrador al final del tema, como un formulario dinámico o una lista interactiva, reforzará los aprendizajes de manera efectiva.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia contextualizando a los aprendedores sobre la evolución de JavaScript, destacando cómo las características modernas de ES6+ han transformado el desarrollo web al hacerlo más eficiente, modular y escalable.
- Es fundamental conectar los conceptos teóricos con ejemplos prácticos y casos de uso real, como la implementación de promesas para manejar asincronía o la utilización de clases en la programación orientada a objetos.
- Las explicaciones deben ir acompañadas de demostraciones en vivo utilizando herramientas de desarrollo como el navegador y un editor de código, para que los aprendedores observen de forma directa cómo las características se implementan y ejecutan. Para mantener el interés de los aprendedores, se recomienda estructurar la sesión en bloques cortos que incluyan teoría, ejemplos prácticos y actividades de codificación guiada, promoviendo la participación.
- Presta especial atención a los conceptos relacionados con asincronía (promesas, `async/await`) y el paradigma de programación orientada a objetos, ya que estos suelen generar dudas debido a su naturaleza abstracta y su integración con JavaScript, un lenguaje originalmente basado en prototipos. Para aclarar estas dudas, se pueden emplear analogías sencillas, diagramas visuales y ejercicios prácticos como la creación de funciones asincrónicas o la modelación de objetos utilizando clases.
- Explica de forma clara los conceptos como el *spread operator*, la desestructuración y el manejo del contexto con funciones flecha deben ser explicados con ejemplos claros, comparando su uso frente a métodos tradicionales.

Recomendaciones: utiliza plataformas interactivas como CodePen o Replit para que los aprendedores puedan probar el código en tiempo real, fomentando una comprensión más profunda y colaborativa.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia contextualizando a los aprendedores sobre la importancia de estos frameworks en el desarrollo web moderno. Es esencial enfatizar cómo han transformado la construcción de interfaces de usuario al simplificar tareas complejas, mejorar el rendimiento y facilitar la reutilización de componentes.
- Utiliza ejemplos prácticos y visuales para explicar las características clave de React, Angular y Vue, destacando sus diferencias y escenarios de uso. Durante la explicación, es crucial desglosar conceptos como el Virtual DOM, los componentes reutilizables, el flujo de datos unidireccional y la programación reactiva, utilizando diagramas y simulaciones en vivo para ilustrar su funcionamiento. Estos recursos ayudarán a los aprendedores a conectar la teoría con la práctica.
- Es probable que existan dudas en la diferencia entre bibliotecas y frameworks, el concepto de Virtual DOM y cómo elegir el framework adecuado para un proyecto. Para aclarar estas inquietudes, proporcionar ejemplos comparativos concretos y resuelve casos prácticos en clase.

- Profundiza en los conceptos de modularidad, inyección de dependencias y gestión del estado, explicando su relevancia en proyectos reales.
- Utiliza analogías como comparar el virtual DOM con un "borrador eficiente antes de imprimir" puede facilitar la asimilación de temas complejos. Por último, se recomienda resaltar la importancia de la documentación oficial como recurso para el aprendizaje continuo.

Recomendaciones: se recomienda hacer una práctica, por ejemplo, que integre conceptos fundamentales de frontend con el backend. La tarea consistirá en **crear un cliente web para consumir la API RESTful desarrollada previamente**, utilizando uno de los frameworks de frontend abordados (React, Angular o Vue). El objetivo sería construir una interfaz funcional que permita interactuar con la API y realizar las operaciones de CRUD sobre las tareas (*to-do list*).

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

Al profesor impartidor, se le recomienda lo siguiente:

- Aborda el tema enfatizando su relevancia en la creación de productos digitales exitosos. Es crucial iniciar estableciendo la diferencia entre UI y UX, ilustrando cómo la interfaz (UI) se enfoca en los aspectos visuales e interactivos, mientras que la experiencia (UX) se centra en la percepción y satisfacción del usuario durante la interacción.
- Utiliza ejemplos prácticos de la vida cotidiana, como aplicaciones populares, para que los aprendedores puedan identificar cómo se implementan estos conceptos.
- Es probable que las dudas más frecuentes pueden surgir en la relación entre los principios estéticos y funcionales de UI y la conexión entre usabilidad y satisfacción del usuario en UX. Estas inquietudes se deben aclarar con ejemplos gráficos y análisis de casos, destacando cómo un mal diseño puede impactar negativamente la experiencia general del usuario. Se recomienda emplear herramientas como *wireframes* o prototipos básicos en clase para fomentar el aprendizaje práctico.
- Explica de forma clara los principios de diseño de UI y las características de UX, ya que estos forman la base de un producto funcional y atractivo. Además, aborda cómo la jerarquía visual, la accesibilidad y la consistencia influyen en UI, mientras que en UX se deben profundizar temas como la usabilidad y el enfoque en el usuario

Recomendación: realiza dinámicas interactivas donde los aprendedores evalúen aplicaciones o sitios web reales, identificando fortalezas y debilidades en términos de UI/UX. Además, organiza simulaciones en las que los aprendedores diseñen una interfaz simple considerando principios clave, para consolidar los conceptos mediante aprendizaje experiencial.

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

Al profesor impartidor, se le recomienda lo siguiente:

- El tema de "Desarrollo backend con Node.js" se debe abordar destacando su relevancia como tecnología asíncrona y no bloqueante, ideal para aplicaciones modernas que requieren escalabilidad y alta capacidad de respuesta.
- Inicia la clase con un panorama general del funcionamiento del **Event Loop**, utilizando diagramas y simulaciones para explicar cómo gestiona múltiples solicitudes de manera

eficiente. Posteriormente, se debe realizar una introducción práctica con un ejemplo básico de creación de un servidor utilizando el módulo `http`, lo que permitirá a los aprendedores comprender su funcionamiento de manera aplicada.

- Es crucial intercalar teoría con práctica, mostrando ejemplos concretos del uso de *callbacks*, promesas y la sintaxis `async/await`, ya que estas son herramientas clave para manejar la asincronía en Node.js. Para ello, utiliza entornos de desarrollo como Visual Studio Code y herramientas de debugging nativas de Node.js o Chrome DevTools para demostrar cómo depurar aplicaciones en tiempo real.
- Los conceptos que podrían generar mayor confusión son el Event Loop, la diferencia entre programación síncrona y asíncrona, y el manejo adecuado de errores. Para aclarar estas dudas, se sugiere emplear analogías simples y ejemplos visuales que conecten estas ideas con situaciones cotidianas, como filas en un restaurante (Event Loop).
- Es fundamental enfatizar cómo manejar los errores en los diferentes enfoques de asincronía (*callbacks*, promesas y `async/await`), muestra tanto buenas prácticas como los errores comunes que se deben evitar. Finalmente, dedica tiempo a la explicación de los módulos `fs` y `http`, resaltando cómo estos simplifican la interacción con archivos y la creación de servidores.

Recomendación: fomenta la participación mediante pequeños desafíos prácticos que los aprendedores resuelvan en parejas, promoviendo el aprendizaje colaborativo y asegurándose de que comprendan cada concepto antes de avanzar al siguiente.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia contextualizando la importancia del framework dentro del desarrollo full stack, destacando cómo simplifica y organiza la construcción de aplicaciones web mediante rutas, middleware y modularización.
- Utiliza un ejemplo práctico y funcional, como la creación de una ruta básica que responda con "Hola Mundo". Este ejemplo permitirá que los aprendedores comprendan rápidamente los conceptos clave de Express.js, como la estructura de una aplicación y el uso de métodos HTTP.
- Explica la importancia de la modularización, utilizando ejemplos con archivos separados como **index.js** y **hello.js**, para reforzar buenas prácticas de desarrollo. Prioriza las demostraciones en vivo y el uso de herramientas de depuración como **console.log** para observar el flujo de datos en tiempo real.
- Es probable que las dudas se centren en el manejo de middleware y la implementación de rutas dinámicas, especialmente en proyectos más complejos. Para aclararlas, se sugiere realizar una comparación visual entre una ruta estática y una dinámica, demostrando cómo los parámetros como `:userId` interactúan con las peticiones. Así mismo desglosa ejemplos de middleware personalizados, explicando el flujo de ejecución con diagramas de secuencia y destacando el uso de la función `next()`.
- Explica de forma clara los conceptos: el objeto `req` y `res`, la modularización del código mediante el router y el manejo eficiente de respuestas con códigos de estado HTTP y formato JSON.

Recomendaciones: emplea una analogía de mensajería, donde cada ruta es un buzón y el middleware son filtros que verifican los mensajes antes de ser entregados; esto puede facilitar la comprensión de estos temas técnicos. Además, usa ejercicios prácticos, como la creación de una API básica, ayudará a consolidar los conocimientos adquiridos.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia contextualizando la importancia de las bases de datos en el desarrollo de aplicaciones web modernas, haciendo énfasis en su papel esencial en la gestión de información.
- Compara las características principales de las bases de datos SQL y NoSQL mediante ejemplos prácticos, como el caso de una tienda en línea, para ilustrar cómo se adaptan a diferentes necesidades. Posteriormente, detalla las operaciones CRUD, explicando su relevancia en la manipulación de datos y destacando cómo se implementan en el backend mediante ejemplos claros, utilizando frameworks y herramientas populares como Express.js con MongoDB o MySQL.
- Es crucial que las explicaciones incluyan tanto el enfoque teórico como práctico, fomentando la participación de los aprendedores con ejercicios como diseñar *endpoints* CRUD para diferentes modelos de datos.
- Las dudas más frecuentes pueden surgir en la elección entre bases de datos SQL y NoSQL y en la comprensión de cómo implementar operaciones CRUD en distintos entornos. Para aclararlas, se deben utilizar ejemplos visuales y comparativos, como diagramas que ilustren la estructura de una base de datos SQL frente a un documento JSON de MongoDB.
- Es recomendable realizar actividades prácticas donde los aprendedores diseñen y prueben consultas o endpoints. Los conceptos que requieren mayor detalle son las claves primarias y foráneas en SQL, la escalabilidad horizontal en NoSQL y el manejo de errores en operaciones CRUD

Recomendaciones: integra simulaciones interactivas en tiempo real para reforzar la comprensión. Por ejemplo, diseñar junto con los aprendedores una pequeña API funcional desde cero utilizando herramientas como Postman para probar las operaciones CRUD en un entorno realista. Durante la simulación, plantea situaciones inesperadas como errores en las consultas, conflictos de datos o problemas de escalabilidad, permitiendo que los aprendedores resuelvan los retos. Este enfoque no solo fortalece la habilidad técnica, sino que fomenta el aprendizaje colaborativo y la toma de decisiones en contextos reales.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Inicia destacando la relevancia de la seguridad en el desarrollo de aplicaciones web, relacionando el contenido con escenarios cotidianos que los aprendedores reconozcan, como el acceso a servicios de *streaming* o redes sociales.
- Presenta una introducción clara de los conceptos de autenticación y autorización, diferenciándolos desde el principio para evitar confusiones. La explicación debe incluir

ejemplos prácticos del uso de JWT y OAuth, enfatizando sus aplicaciones reales, como la autenticación mediante Google o la generación de *tokens* para APIs. Durante la sesión, fomenta la participación, solicitando a los aprendedores que analicen fragmentos de código proporcionados, explicando las funciones de cada componente y cómo contribuyen al flujo de seguridad.

- Es probable que pueda existir dudas en la diferencia entre la codificación, cifrado y la firma digital en JWT, así como el flujo detallado de OAuth al delegar la autorización a servicios de terceros. Para aclarar estas dudas, se recomienda utilizar diagramas que representen gráficamente los procesos y esquemas de comparación que diferencien términos relacionados. Además, explica con especial detalle el uso de variables de entorno y la importancia de limitar la duración de los tokens para reforzar la seguridad.

Recomendaciones: integra una simulación gamificada utilizando un entorno de ciberseguridad controlado, como una plataforma tipo "*Capture the Flag*" (CTF), en la que los aprendedores deban proteger una aplicación de posibles vulnerabilidades relacionadas con autenticación y autorización. Puedes crear escenarios donde los aprendedores, por equipos o individual, deban implementar soluciones para proteger tokens JWT contra ataques de fuerza bruta o interceptar y mitigar riesgos de sesión en un flujo OAuth mal configurado. A través de esta dinámica, los aprendedores no solo aplican los conceptos de forma práctica, sino que también desarrollan habilidades críticas para identificar y solucionar problemas de seguridad en tiempo real, simulando un entorno profesional y competitivo que los motive a explorar soluciones más allá del aula.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Comienza explicando los conceptos básicos de una API como interfaz de comunicación entre sistemas, destacando su papel en la arquitectura cliente-servidor. El enfoque inicial se debe centrar en cómo las RESTful APIs utilizan métodos HTTP para gestionar operaciones CRUD, explicando su modularidad y escalabilidad.
- Presenta ejemplos prácticos con endpoints claros que utilicen URLs en plural, manejo de recursos, y códigos de estado HTTP estándar.
- Aborda GraphQL como una herramienta alternativa, enfatizando sus ventajas al optimizar consultas complejas y permitir un control detallado sobre los datos solicitados. Se recomienda integrar comparativas claras entre REST y GraphQL, usando casos prácticos, por ejemplo una red social para compartir fotos, para resaltar sus diferencias y aplicaciones específicas.
- Las principales dudas pueden surgir al comprender la abstracción entre frameworks (como Express.js) y arquitecturas (como REST o GraphQL) y al identificar en qué escenarios cada tecnología es más adecuada. Para aclararlas, se recomienda utilizar analogías sencillas que ilustren la flexibilidad de REST y la especificidad de GraphQL, reforzando el aprendizaje con diagramas de flujo y ejemplos de código prácticos.
- Los conceptos que requieren mayor detalle incluyen la definición de recursos en REST, la estructura de consultas y mutaciones en GraphQL, y el uso de *resolvers* para gestionar datos.

Recomendaciones: utiliza herramientas interactivas como Postman o Apollo Studio para realizar demostraciones en tiempo real, permitiendo a los aprendedores visualizar cómo las solicitudes y respuestas se gestionan en ambos paradigmas. Además, utiliza ejercicios para fomentar el aprendizaje práctico sobre el diseño y consumo de APIs, las cuales se

pueden estructurar en varias etapas, abordando tanto REST como GraphQL, para que los aprendedores desarrollen habilidades completas.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Comienza con una explicación clara de cómo la nube ha transformado los modelos tradicionales de despliegue de aplicaciones, enfatizando su flexibilidad, escalabilidad y eficiencia.
- Estructura la enseñanza en tres partes: introducción al concepto de infraestructura en la nube, explicación del uso de contenedores y la importancia de herramientas de orquestación como Kubernetes. La ejemplificación, como el caso de Netflix, puede ser utilizada para contextualizar los beneficios reales y tangibles en el ámbito empresarial.
- Durante la sesión, se pueden emplear esquemas visuales y diagramas de flujos para describir procesos como CI/CD, despliegue continuo y gestión con contenedores. Es fundamental relacionar cada subtema con aplicaciones prácticas para mantener el interés y facilitar la comprensión de los conceptos.
- Es probable que los aprendedores puedan tener dudas sobre los aspectos técnicos de los contenedores y su orquestación, particularmente en las configuraciones iniciales de Docker y Kubernetes. Para aclarar estas dudas, se sugiere realizar una demostración práctica en tiempo real sobre la creación de un contenedor a partir de un Dockerfile y el despliegue en un entorno local.
- Los conceptos de CI/CD deben ser explicados con ejemplos sencillos que muestren la interacción entre pruebas automatizadas, integración y despliegue. Es importante profundizar en temas como el uso de archivos `.dockerignore`, imágenes ligeras y los parámetros de configuración de Kubernetes, ya que su correcta aplicación es crucial para optimizar los proyectos.

Recomendaciones: incorpora un ejercicio gamificado llamado **"El reto del despliegue óptimo"**. Divide a los aprendedores en equipos y proporciona un escenario ficticio en el que una empresa requiere escalar rápidamente su aplicación debido a un aumento inesperado en la demanda. Cada equipo deberá utilizar herramientas como Docker y una plataforma gratuita (Docker Desktop o Minikube) para diseñar una solución de despliegue en contenedores que cumpla con los requisitos de escalabilidad y eficiencia. Durante el ejercicio, introduce "eventos sorpresa" (como un aumento de tráfico o un fallo en un contenedor) para que los equipos ajusten su solución usando técnicas de orquestación. Este enfoque no solo fortalece la comprensión práctica, sino que también fomenta habilidades de colaboración y resolución de problemas en tiempo real.

- **Por otro lado, revisa y aborda un ejercicio** de implementación de un microservicio Node.js en la nube con Docker y Kubernetes para que lo puedan ver los aprendedores:

Este ejercicio práctico tiene como objetivo mostrar a los aprendedores cómo desplegar un microservicio desarrollado en Node.js utilizando Docker y Kubernetes.

Inicia explicando el propósito del ejercicio: demostrar la creación de un contenedor Docker para un microservicio básico, construir una imagen de este contenedor y, finalmente, desplegarlo en un clúster de Kubernetes. Es recomendable mostrar cada paso en tiempo real, utilizando un entorno local (como Docker Desktop o Minikube) y destacando la relación entre los comandos, los archivos de configuración y el despliegue en la nube.

- **Paso 1. Crear un proyecto Node.js:**

Explica cómo inicializar el proyecto usando `npm init` y cómo instalar Express como dependencia para crear un servidor web básico. Señala la importancia de especificar un puerto configurable en el código (`app.js`) mediante `process.env.PORT`. Esto permite la integración fluida con los contenedores y Kubernetes.

```
1
2 # Inicializar un proyecto Node.js
3 npm init -y
4
5 # Instalar Express como dependencia
6 npm install express
7 |
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- **Paso 2. Código del microservicio (app.js):**

Presenta el código del archivo `app.js`, explicando que este representa un servidor básico de Node.js y que responde con un mensaje desde un contenedor Docker. Es importante enfatizar la estructura del código.

```
1
2 // app.js
3 const express = require('express');
4 const app = express();
5
6 // Configuración del puerto
7 const PORT = process.env.PORT || 4000;
8
9 // Ruta básica
10 app.get('/', (req, res) => {
11   res.send('¡Hola desde un contenedor Docker!');
12 });
13
14 // Iniciar el servidor
15 app.listen(PORT, () => {
16   console.log(`Servidor corriendo en el puerto ${PORT}`);
17 });
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- **Paso 3. Crear un archivo Dockerfile:**

Muestra cómo construir un archivo Dockerfile optimizado utilizando una imagen base ligera (`node:16-alpine3.14`). Destaca el uso de comandos como `WORKDIR`, `COPY`, `RUN npm install` y la exposición del puerto `4000`. Explica que el Dockerfile empaqueta el microservicio con todas sus dependencias, asegurando portabilidad entre entornos.

```
1
2 # Usar una imagen base ligera de Node.js
3 FROM node:16-alpine3.14
4
5 # Crear un directorio de trabajo
6 WORKDIR /app
7
8 # Copiar las dependencias del proyecto
9 COPY package.json .
10
11 # Instalar las dependencias
12 RUN npm install
13
14 # Copiar el resto de los archivos al contenedor
15 COPY . .
16
17 # Exponer el puerto de la aplicación
18 EXPOSE 4000
19
20 # Comando para ejecutar la aplicación
21 CMD ["node", "app.js"]
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- **Paso 4. Crear archivo .dockerignore:**

Resalta la utilidad de este archivo para excluir elementos innecesarios (como `node_modules`) en el proceso de construcción, optimizando la imagen y reduciendo el tiempo de despliegue.

```
1
2 node_modules
3 npm-debug.log
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- **Paso 5. Construir y ejecutar la imagen Docker:**

Guía el proceso de construcción de la imagen Docker con el comando `docker build`, asignándole un nombre identificable (`microservicio-node`). Luego, muestra cómo ejecutar el contenedor en segundo plano (`docker run -d`) mapeando el puerto local `8080` al puerto `4000` del contenedor.

```
1
2 # Construir la imagen Docker
3 docker build -t microservicio-node .
4
5 # Ejecutar el contenedor
6 docker run -d -p 8080:4000 microservicio-node
7
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- **Paso 6. Crear un manifiesto para Kubernetes (deployment.yaml):**

Explica el contenido del archivo `deployment.yaml`, destaca la especificación de réplicas, contenedores y puertos. Esto permite que el clúster escale automáticamente según la carga.

```
1
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: microservicio-node
6 spec:
7   replicas: 2
8   selector:
9     matchLabels:
10      app: microservicio-node
11 template:
12   metadata:
13     labels:
14       app: microservicio-node
15   spec:
16     containers:
17       - name: microservicio-node
18         image: microservicio-node
19         ports:
20           - containerPort: 4000
21 ---
22 apiVersion: v1
23 kind: Service
24 metadata:
25   name: microservicio-node-service
26 spec:
27   selector:
28     app: microservicio-node
29   ports:
30     - protocol: TCP
31       port: 80
32       targetPort: 4000
33   type: LoadBalancer
34
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- **Paso 7. Desplegar en Kubernetes:**

Finaliza mostrando cómo aplicar el manifiesto (`kubectl apply -f deployment.yaml`) y verifica los recursos creados con `kubectl get all`. Señala cómo acceder al microservicio desplegado mediante el *Load Balancer* o puerto configurado.

```
1
2 # Crear el deployment y servicio en Kubernetes
3 kubectl apply -f deployment.yaml
4
5 # Verificar los recursos creados
6 kubectl get all
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

El microservicio ahora está desplegado en un clúster de Kubernetes. Puedes acceder a él a través del Load Balancer o el puerto configurado.

Recomendación del ejemplo: realiza un "debug en vivo" durante la clase. Crea intencionalmente un pequeño error en el manifiesto de Kubernetes (por ejemplo, un nombre de imagen incorrecto) y muestra a los aprendedores cómo identificar y corregir errores utilizando comandos como `kubectl logs` o `kubectl describe pod`. Este enfoque práctico les ayudará a entender el proceso de resolución de problemas en entornos reales.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Enfatiza la relación entre los casos de prueba y la calidad del software, así como su papel en el ciclo de vida del desarrollo. Se sugiere iniciar la sesión con una explicación clara sobre la estructura de un caso de prueba, ilustrando su propósito con ejemplos prácticos que incluyan identificadores, pasos y resultados esperados frente a resultados actuales.
- Es importante subrayar cómo una correcta planificación y ejecución de pruebas contribuye a detectar fallos de manera temprana. A lo largo de la clase, se deben fomentar discusiones en torno a la importancia de una nomenclatura estándar para los casos de prueba y cómo esta facilita el trabajo en equipo y la escalabilidad de proyectos.
- Es probable que las dudas más comunes puedan surgir al identificar errores en los resultados actuales frente a los esperados; se recomienda aclarar estas dudas mediante el análisis conjunto de ejemplos adicionales y el uso de herramientas como Jira para gestionar casos de prueba.
- Al abordar la depuración, es esencial destacar su rol continuo en el desarrollo y la diferencia clave con los casos de prueba, que están más orientados a la validación de funcionalidades. Se sugiere incluir una demostración práctica del uso de `logs` y depuradores, enfatizando cómo los mensajes claros y las herramientas adecuadas, como Chrome DevTools, contribuyen a localizar y solucionar errores.

Recomendaciones: implementa un **laboratorio de errores gamificado**, donde los aprendedores, organizados en equipos, compitan para identificar y corregir fallos predefinidos en un entorno controlado. Cada equipo recibiría un proyecto con errores simulados que varíen en complejidad, y puntos se otorgarían por encontrar y solucionar errores correctamente en el menor tiempo posible. Esto no solo fomentará la práctica en depuración, sino que también incentivará el aprendizaje dinámico y colaborativo. Para enriquecer la experiencia, se puede usar software de depuración en tiempo real y permitir

que los equipos documenten sus procesos en un tablero compartido, lo que refuerza habilidades de trabajo en equipo y comunicación técnica.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Destaca las similitudes y diferencias entre el desarrollo móvil y el desarrollo web previamente estudiado en el curso. Inicia conectando los conocimientos previos de los aprendedores sobre la gestión de APIs, manejo del estado y componentes reutilizables, para luego introducir las características específicas del desarrollo móvil, como la distinción entre aplicaciones nativas e híbridas.
- Enfatiza en cómo frameworks como React Native combinan conceptos de desarrollo web con la flexibilidad para adaptarse a diferentes plataformas móviles, permitiendo así un flujo de aprendizaje más fluido. La configuración del entorno de desarrollo se debe explicar mediante una demostración práctica que muestre paso a paso cómo instalar y configurar las herramientas esenciales (Node.js, Android Studio, Xcode), acompañada de una discusión sobre las limitaciones de los sistemas operativos.
- Las dudas más frecuentes pueden surgir al comprender la diferencia entre componentes nativos y personalizados, así como al implementar la navegación con React Navigation y las personalizaciones según el sistema operativo mediante Platform y Native Modules. Estas dudas se pueden aclarar mostrando ejemplos comparativos entre el desarrollo web con React y el desarrollo móvil con React Native, utilizando fragmentos de código que ilustren claramente los conceptos.
- Es crucial detallar cómo funcionan las rutas y la navegación entre pantallas mediante *stacks*, resaltando su similitud con el historial de navegación web. También enfócate en explicar de forma detallada los pasos para la inicialización de un proyecto, la instalación de dependencias y los procesos de despliegue en iOS y Android.

Recomendaciones: implementa un **“debugging colaborativo en vivo”** como parte de la clase. Presenta un error intencional en un fragmento de código React Native, como una navegación mal configurada o un componente con estilos incorrectos, y solicita a los aprendedores identificar y solucionar el problema en tiempo real utilizando un simulador compartido en pantalla. Este enfoque no solo refuerza la habilidad para detectar y corregir errores, sino que también fomenta el pensamiento crítico y el trabajo en equipo, simulando un entorno de desarrollo profesional. Además, esta práctica ayuda a los aprendedores a familiarizarse con las herramientas de debugging disponibles en los IDEs utilizados.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Guía a los aprendedores enfatizando la importancia de dividir el desarrollo en etapas claras para facilitar el cumplimiento de cada parte de la actividad.
- Dedicar tiempo para mostrar ejemplos prácticos de cómo estructurar correctamente el código HTML y aplicar estilos con CSS. Un consejo clave es realizar una demostración breve de la creación de la carpeta del proyecto y los archivos **index.html** y **styles.css** en el entorno de desarrollo sugerido, como Visual Studio Code. Esto ayudará a evitar confusiones iniciales y establecerá un estándar de organización desde el comienzo.

- Se sugiere recalcar la importancia de la semántica en el HTML, explicando cómo utilizar etiquetas como `<header>`, `<section>` y `<footer>` de manera adecuada para mantener una estructura lógica y profesional.
- Durante la estilización y diseño responsivo, facilita la comprensión al presentar ejemplos de **media queries** y su impacto en el diseño para diferentes tamaños de pantalla. Es útil proporcionar ejemplos visuales comparativos que muestren la diferencia entre un diseño estático y uno responsivo. Puedes sugerir herramientas en línea como Chrome DevTools para que los aprendedores prueben y ajusten su diseño responsivo en tiempo real.
- Asigna tiempo para revisar y validar el formulario de contacto, enfocándote en las etiquetas de accesibilidad como *label* y atributos como *alt* en imágenes, explicando cómo estos mejoran la experiencia del usuario.
- **Recomendaciones:** puedes sugerir herramientas en línea como Chrome DevTools para que los aprendedores prueben y ajusten su diseño responsivo en tiempo real. Organiza sesiones cortas de revisión grupal donde los aprendedores muestren sus avances y reciban retroalimentación inmediata, fomentando así el aprendizaje colaborativo.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Presta especial atención a las instrucciones relacionadas con la manipulación del DOM y el uso de programación orientada a objetos (OOP), ya que son los aspectos más críticos y probablemente donde los aprendedores encuentren mayor dificultad.
- Debes estar atento a la implementación de la clase Tarea y su integración en la clase GestorDeTareas, así como al correcto manejo de eventos en los botones para agregar, editar y eliminar tareas. Estas secciones demandan un entendimiento profundo de cómo se estructura y manipula el DOM, además de aplicar características modernas de ES6+ como funciones flecha, *let*, *const* y *template literals*, lo cual podría generar dudas en los aprendedores menos familiarizados con estas herramientas.
- Se recomienda hacer demostraciones prácticas, para enseñar el paso a paso sobre cómo se construyen y manipulan los elementos del DOM y cómo se implementan las clases con métodos para manejar las tareas.
- Se sugiere animar a los aprendedores a plantear sus dudas en un entorno colaborativo y aprovechar la revisión de código en grupo para identificar errores comunes.
- Proporciona ejemplos claros de cómo implementar LocalStorage, ya que esto será clave para garantizar que los aprendedores logren completar con éxito el desafío adicional.

Recomendaciones: divide la actividad en etapas más manejables, verificando el progreso de los aprendedores en cada paso antes de avanzar al siguiente. Por ejemplo, asegurarse de que comprendan y ejecuten correctamente la adición de tareas antes de proceder con la funcionalidad de edición y eliminación.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Presta atención a las instrucciones relacionadas con la configuración inicial del proyecto y la implementación de la autenticación y sesiones. Estas fases, detalladas en los apartados a, e y f, son fundamentales para garantizar que la API RESTful funcione correctamente y cumpla con los requisitos de seguridad y funcionalidad.
- La configuración de las dependencias, el manejo de datos con el módulo `fs` y el desarrollo de middleware para errores son áreas donde los aprendedores podrían enfrentar mayores retos debido a la necesidad de comprender conceptos como la asincronía y el manejo seguro de información.
- Durante la actividad, es recomendable verificar que los aprendedores hayan entendido cómo utilizar herramientas como `console.log` y la opción `--inspect` de Node.js para depuración, ya que esto facilitará la identificación de problemas en su implementación.
- Fomenta el uso de herramientas de inteligencia artificial (IA) para obtener ejemplos prácticos de código, como se menciona en las instrucciones, guiando a los aprendedores para que evalúen críticamente estas soluciones y las adapten a los requerimientos específicos de la actividad.
- Proporcionar ejemplos claros y realiza revisiones en tiempo real de los archivos clave, como `server.js` y `tareas.json`

Recomendaciones: para resolver dudas comunes, como la protección de rutas y la integración de middleware personalizado, se recomienda organizar una sesión breve de preguntas y respuestas. Asimismo, enfatiza la importancia de documentar y presentar el funcionamiento de la API ayudará a los aprendedores a estructurar adecuadamente su entrega final.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Presta atención a las instrucciones relacionadas con la configuración del servidor y la implementación del flujo de autenticación y autorización mediante JWT. Estas tareas suelen ser complejas para los aprendedores, especialmente si tienen experiencia limitada en la integración de middleware y la gestión de tokens de seguridad.
- Es esencial guiar a los aprendedores en la generación de pruebas unitarias utilizando Jest y herramientas de IA como Copilot, ya que esta parte puede generar dudas sobre cómo ajustar correctamente las pruebas generadas automáticamente para que cumplan con los requisitos específicos del proyecto. Durante estas etapas, se recomienda proporcionar ejemplos prácticos y aclarar cualquier duda técnica a través de sesiones interactivas donde se resuelvan problemas en tiempo real.
- Explica claramente la importancia de documentar los requerimientos funcionales y no funcionales, así como la justificación de las decisiones de diseño, como la elección de la plataforma SaaS para el despliegue.
- Fomenta el uso colaborativo de plataformas como GitHub para el manejo de repositorios; esto garantizará que los aprendedores desarrollen competencias relevantes para proyectos reales.

Recomendaciones: divide la actividad en etapas bien definidas y establecer hitos claros, revisando continuamente el avance de los aprendedores en cada fase. Se sugiere utilizar

diagramas y ejemplos visuales para explicar conceptos clave como el pipeline de CI/CD y los diagramas de entidad-relación.

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

Al profesor impartidor, se le recomienda lo siguiente:

- Presta especial atención a la primera y segunda instrucción, ya que estas requieren que los aprendedores creen sus cuentas en las plataformas IBM Skillbuild y Credly. Es importante asegurarse de que todos los aprendedores hayan completado correctamente el registro y comprendan cómo navegar por la ruta de aprendizaje.
- Las primeras interacciones con la plataforma pueden generar dudas, como problemas de acceso, pérdida de contraseñas o falta de claridad en los pasos a seguir. Una manera efectiva de aclarar estas dudas es dedicar tiempo al inicio de la actividad para realizar una demostración guiada del registro y navegación en IBM Skillbuild, mostrando paso a paso cómo inscribirse en los cursos, acceder a los materiales y visualizar las evaluaciones.
- Las instrucciones tres y cuatro también podrían generar confusión, especialmente en lo relacionado con la creación de la cuenta en Credly y los pasos para obtener la certificación. Se recomienda que el impartido haga una breve sesión de preguntas y respuestas. Después de explicar estas indicaciones, proporciona capturas de pantalla o guías visuales para que los aprendedores puedan consultarlas de manera independiente.

Recomendaciones: fomenta el uso de un calendario o cronograma que permita a los aprendedores estar organizados para completar los cursos y evaluaciones dentro del tiempo asignado. Finalmente, asegúrate que los aprendedores entiendan la importancia de alcanzar la calificación mínima requerida (80%) para obtener la certificación, reforzando la motivación con ejemplos prácticos de los beneficios de esta credencial en el ámbito profesional.

Notas para el profesor impartidor. Estas corresponden a la explicación del avance del reto (fase I):

Al profesor impartidor, se le recomienda lo siguiente:

- Presta especial atención a las instrucciones relacionadas con la configuración inicial del proyecto y la implementación del backend.
- La instalación de dependencias como `express`, `jsonwebtoken` y herramientas específicas para bases de datos puede generar dudas, especialmente en la configuración de variables de entorno mediante `dotenv`. Se sugiere utilizar ejemplos prácticos, como estructurar un archivo `.env` para almacenar credenciales sensibles y cómo integrarlo correctamente en el proyecto.
- La implementación de autenticación básica con JWT y middleware personalizado puede ser un punto crítico. Por lo que, debes estar preparado para aclarar conceptos relacionados con el manejo de tokens y su validación en las rutas protegidas. Es recomendable demostrar ejemplos básicos y realizar pruebas con herramientas como Postman para garantizar el entendimiento de los aprendedores.
- Guía a los aprendedores en la creación de un diseño básico que cumpla con principios fundamentales de UI/UX, destacando la importancia de CSS y JavaScript en la interacción dinámica del usuario.

- Promueve el uso de herramientas de inteligencia artificial para generar y evaluar plantillas de diseño en CSS o sugerencias de código, fomentando un enfoque crítico y reflexivo hacia las propuestas generadas. Asegúrate que los aprendedores realicen pruebas continuas y utilicen GitHub para el control de versiones; esto facilitará la detección y corrección de errores en las distintas etapas del reto.

Recomendaciones: orienta a los aprendedores a elegir un enfoque claro y práctico para su aplicación web, sugiriendo ejemplos de proyectos que sean relevantes y de interés, como un sistema de gestión de tareas, una plataforma de reservas en línea, o una tienda virtual básica.

Se recomienda enfatizar que el diseño del backend se debe centrar en operaciones CRUD funcionales y la implementación de autenticación mediante JWT, asegurando que cada módulo sea probado con herramientas como Postman. Para el frontend, se puede sugerir la creación de una interfaz simple pero funcional que incluya una página de inicio de sesión y una vista principal con datos dinámicos, destacando la importancia de aplicar principios básicos de diseño UI/UX con CSS y JavaScript para mejorar la experiencia del usuario. Finalmente, fomentar el uso de inteligencia artificial para generar ideas o plantillas puede inspirar creatividad y optimizar el tiempo de desarrollo.

Notas para el profesor impartidor. Estas corresponden a la explicación de la entrega final del reto (fase II):

Al profesor impartidor, se le recomienda lo siguiente:

- Debes poner atención en las instrucciones relacionadas con la implementación de la autenticación y autorización segura utilizando OAuth o JWT, así como la integración de una API externa en el proyecto. Estas actividades suelen ser desafiantes para los aprendedores, especialmente en lo que respecta a configurar permisos específicos en roles de usuario y manejar adecuadamente las respuestas y errores provenientes de APIs externas.
- Se recomienda dedicar tiempo a explicar cómo las herramientas de inteligencia artificial (IA) pueden proporcionar ejemplos prácticos, como la implementación de paginación o el consumo de APIs, y cómo evaluar y adaptar esos ejemplos a los requerimientos específicos del proyecto. Asimismo, se debe aclarar la importancia del middleware personalizado para gestionar errores y facilitar el debugging en el backend, puntos que pueden generar dudas debido a su nivel técnico.
- Enfatiza la importancia de la documentación técnica avanzada y su relación con el éxito del proyecto. Los aprendedores podrían necesitar orientación adicional en la redacción y organización de dicha documentación, por lo que es recomendable proporcionar plantillas o ejemplos previos.
- Se recomienda realizar sesiones de retroalimentación grupal o individual para resolver dudas específicas sobre la integración del frontend y backend, el despliegue en la nube, y las pruebas de seguridad.

Recomendaciones: fomenta la colaboración entre los aprendedores mediante actividades de revisión por pares, utilizando herramientas como GitHub para facilitar el intercambio y la corrección conjunta de código. Finalmente, se sugiere que motivar a los aprendedores a realizar pruebas iterativas durante el desarrollo y antes de la entrega, asegurando que cada funcionalidad cumpla con los requisitos del reto.

Rúbrica avance del reto (fase I)

Criterios de evaluación	Nivel de desempeño			%
	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	
1. Estructura del proyecto	15 - 13 El proyecto está completamente organizado, con una estructura clara y coherente de archivos y carpetas para <i>frontend</i> , <i>backend</i> y controladores.	12 - 10 La estructura del proyecto es adecuada, pero algunas carpetas o archivos podrían estar mejor organizados o ser más intuitivos.	9 - 0 La estructura del proyecto es desorganizada o confusa, dificultando la comprensión de las funciones o del flujo de trabajo entre <i>frontend</i> y <i>backend</i> .	15
2. Funcionalidad CRUD	20 - 18 Las operaciones CRUD (<i>create</i> , <i>read</i> , <i>update</i> y <i>delete</i>) funcionan correctamente en todas las rutas, con integración completa a la base de datos y sin errores.	17 - 15 Las operaciones CRUD funcionan en su mayoría, aunque puede haber errores menores en alguna de las rutas o en la integración con la base de datos.	14 - 0 Las operaciones CRUD no funcionan correctamente, o no se ha completado la integración con la base de datos.	20
3. Autenticación básica (JWT)	20 - 18 La autenticación está implementada correctamente utilizando JWT, protegiendo las rutas adecuadas y controlando el acceso según los permisos definidos.	17 - 15 La autenticación JWT está implementada, aunque algunas rutas podrían no estar protegidas o la configuración presenta errores de seguridad o funcionamiento.	14 - 0 No se ha implementado la autenticación JWT, o está mal configurada, lo que permite el acceso sin autorización a rutas protegidas.	20
4. Diseño de interfaz y experiencia de usuario (UI/UX)	15 - 13 La interfaz es visualmente atractiva, fácil de navegar y cumple los principios básicos de UI/UX; incluye interacciones dinámicas y buena manipulación del DOM con JS.	12 - 10 La interfaz es funcional y tiene un diseño adecuado, pero la navegación o el estilo visual se podría mejorar en algunos aspectos; presenta interacciones básicas en JS.	9 - 0 El diseño de la interfaz es pobre, carece de elementos de UI/UX básicos y la interacción es mínima o inexistente; la manipulación del DOM es deficiente o ausente.	15

5. Despliegue preliminar	15 - 13 La aplicación está desplegada correctamente en un entorno local y en la nube (si es posible), mostrando estabilidad y funcionalidad completa en ambos entornos.	12 - 10 La aplicación está desplegada y funcional en un entorno local; el despliegue en la nube tiene problemas menores o no se realizó debido a dificultades técnicas justificadas.	9 - 0 La aplicación no se ha desplegado en ningún entorno o el despliegue falla en aspectos fundamentales, afectando significativamente la funcionalidad.	15
6. Documentación técnica y entregables	15 - 13 La documentación es detallada y clara, describe la estructura del proyecto, los <i>endpoints</i> de la API y el flujo del sistema; todos los entregables se presentan.	12 - 10 La documentación es adecuada, pero puede carecer de detalles o claridad en algunos aspectos; algunos entregables están incompletos o faltan elementos menores.	9 - 0 La documentación es insuficiente o poco clara, dificulta la comprensión del proyecto; además faltan entregables claves o están incompletos.	15
TOTAL				100%

Rúbrica de la entrega final del reto (fase II)

Criterios de evaluación	Nivel de desempeño			%
	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	
1. Funcionalidad completa	20 - 18 La aplicación es completamente funcional, cubre todas las operaciones CRUD, con integración de una API externa y la autenticación es implementada correctamente, garantizando un flujo de trabajo robusto.	17 - 15 La aplicación es funcional en gran medida, pero presenta algunas limitaciones en operaciones CRUD o en la integración de la API externa, o la autenticación implementada.	14 - 0 La aplicación no es funcional en una o más áreas clave (CRUD, autenticación, integración de API externa), afectando gravemente su usabilidad.	20
2. Implementación de OAuth/JWT	20 - 18 La autenticación y autorización mediante OAuth o JWT están implementadas de manera completa, además protege adecuadamente todas las rutas y roles definidos.	17 - 15 La autenticación y autorización mediante OAuth o JWT están implementadas, pero con protecciones parciales en ciertas rutas o roles.	14 - 0 La autenticación y autorización son deficientes o no están implementadas, deja las rutas vulnerables o sin protección.	20
3. Frontend completo con framework	15 - 13 El frontend utiliza efectivamente un framework (React, Angular o Vue.js), con una estructura clara, navegación intuitiva y experiencia de usuario optimizada.	12 - 10 El frontend utiliza un framework con una estructura funcional y navegación adecuada, aunque hay algunas áreas que podrían mejorar en cuanto a usabilidad o experiencia de usuario.	9 - 0 El frontend no utiliza un framework o su implementación es deficiente, afectando la estructura, navegación y experiencia del usuario.	15
4. Despliegue en producción	15 - 13 La aplicación está desplegada correctamente en la nube (Heroku, AWS, DigitalOcean o Render), con URL pública funcional,	12 - 10 La aplicación está desplegada en la nube y es accesible, pero presenta fallas en la integración entre frontend y backend o en su accesibilidad.	9 - 0 La aplicación no está desplegada en la nube o no es accesible públicamente, además presenta fallas	15

	integrando frontend y backend de manera impecable.		significativas en la integración o en el despliegue.	
5. Pruebas y <i>debugging</i>	15 - 13 Se implementan casos de prueba unitarios, funcionales y de seguridad, con un manejo de errores eficiente y una depuración exhaustiva que asegura la estabilidad de la aplicación.	12 - 10 Se implementaron casos de prueba, pero con alcance limitado en pruebas de seguridad o con ciertos errores de manejo que limitan la depuración completa de la aplicación.	9 - 0 No se implementaron pruebas o depuración adecuada, además existen fallas significativas en el manejo de errores que afectan la funcionalidad.	15
6. Creatividad y experiencia de usuario	15 - 13 La aplicación demuestra originalidad en el diseño, con una interfaz intuitiva, es visualmente atractiva y proporciona una excelente experiencia de usuario.	12 - 10 La aplicación tiene un diseño adecuado y proporciona una buena experiencia de usuario, aunque carece de originalidad o podría mejorar en la interfaz y la interacción.	9 - 0 La aplicación no demuestra creatividad ni proporciona una experiencia de usuario adecuada, tiene una interfaz poco intuitiva o es visualmente pobre.	15
TOTAL				100%

Prácticas de bienestar

Práctica 1

Nombre de la práctica	Un momento para respirar.
Descripción de la práctica	Aprender a respirar por la nariz y a tranquilizar tu mente.
Palabras clave	Fortalezas de carácter, autorregulación.
Instrucciones para el aprendizador	<p>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, especialmente, sobre las emociones.</p> <p>En la actualidad vivimos situaciones muy estresantes que provocan que nuestra reacción instintiva y natural ante ellas sea estallar en ira. Pero, las consecuencias de este comportamiento no solo se quedan en nosotros, sino que también pueden llegar a afectar a terceros.</p> <p>A continuación, se presenta un ejercicio que te ayudará a cultivar la fortaleza de autorregulación:</p> <ol style="list-style-type: none"> 1. Toma dos minutos de tu tiempo, siéntate en un lugar cómodo, donde no haya mucho ruido que te pueda distraer. 2. Escucha música de relajación (crea tu propio ambiente de meditación). 3. Comienza a respirar y exhalar por nariz. 4. Trata de que tu respiración y exhalación dure el mismo tiempo. 5. Fija tu mente en tu respiración, en cómo entra y sale el aire de tu cuerpo. <p>Así durante dos minutos.</p> <p>Te recomendamos que si 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 vas aumentando los minutos de este.</p>
Fuente	Conferencia Rosalinda Ballesteros.

Práctica 2

Nombre de la práctica	Fomentando la atención plena.
------------------------------	-------------------------------

Descripción de la práctica	Llevarás a cabo breves ejercicios de meditación para fomentar la atención plena en tus actividades diarias.
Palabras clave	Atención plena, fortalezas de carácter, autorregulación.
Instrucciones para el aprendizador	<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, puedes llevar 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"> 1. Haz tres respiraciones profundas por la nariz y exhala por la nariz. 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? 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? <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>
Fuente	Eby, D. (s.f.). <i>Creativity and Flow Psychology</i> . Recuperado de http://talentdevelop.com/articles/Page8.html

Práctica 03

Nombre de la práctica	Experiencias difíciles.
Descripción de la práctica	En esta práctica podrás analizar las estrategias que seguiste para afrontar problemáticas y cómo aprendiste de tales sucesos.
Palabras clave	Resiliencia.
Instrucciones para el aprendizador	<p>Todos hemos pasado por situaciones complejas, no solo en lo laboral, sino también en el ámbito 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</p>

	<p>resiliencia implica no solo salir adelante después de una situación muy dura, sino incluso 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"> 1. Crea una tabla con tres columnas y cinco filas. 2. En la primera columna escribe un evento difícil o desagradable al que te hayas enfrentado en tu vida. 3. En la segunda columna menciona cuáles son tus creencias sobre esa adversidad. 4. En la tercera columna describe las consecuencias que tiene esa creencia. 5. Cuando termines, lee toda la tabla y reflexiona sobre cómo te ha cambiado cada evento y cómo lo enfrentaste. 6. Escribe al final cómo enfrentarías cada evento hoy en día.
Fuente	<ul style="list-style-type: none"> • Metodología ABC. • Fundamentos de psicología positiva.

Práctica 04

Nombre de la práctica	Concentrarse en lo positivo.
Descripción de la práctica	Analizarás sucesos que te hayan ocurrido recientemente, buscando orientar el análisis hacia las consecuencias positivas.
Palabras clave	Resiliencia y esperanza.
Instrucciones para el aprendiz	<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 automáticamente 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"> 1. Piensa en una noticia reciente que hayas recibido y que creas que es negativa para ti. 2. Luego de analizarla, haz una tabla con tres columnas. En la primera, señala cuál sería el peor de los escenarios posibles que pudieran 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. 3. Reflexiona sobre los tres escenarios, ¿cómo enfrentarías cada uno de ellos? <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>

Fuente	Seligman, M. (2011). <i>Building Resilience</i> . Recuperado de https://hbr.org/2011/04/building-resilience
---------------	---

Práctica 05

Nombre de la práctica	Crecimiento postraumático.
Descripción de la práctica	En esta práctica harás un recuento de las situaciones difíciles a las que te has enfrentado y reflexionarás sobre lo positivo que surgió de ellas.
Palabras clave	Resiliencia.
Instrucciones para el aprendizador	<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 no solo salir adelante después de una situación muy dura, sino incluso 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"> 1. Escribe acerca de un momento en el que enfrentaste una adversidad significativa o pérdida. 2. Primero escribe acerca de las puertas que se te cerraron debido a esa adversidad o pérdida, ¿qué perdiste? 3. Después escribe acerca de las puertas que se abrieron al término o como secuela de esa adversidad o pérdida. 4. ¿Hay nuevas maneras de actuar, pensar o relacionarse que son más probables de suceder ahora?
Fuente	<ul style="list-style-type: none"> • 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

Nombre de la práctica	La mejor versión de ti mismo.
Descripción de la práctica	Escribe acerca de la mejor versión posible de ti mismo durante al menos 20 minutos.
Palabras clave	Emociones positivas, fortalezas de carácter, autorregulación y esperanza.
Instrucciones para el aprendizador	<p>Imagina que dentro de 20 años 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"> • ¿Cómo es esa mejor versión de ti mismo? • ¿Qué hace él o ella cotidianamente? • ¿Qué dicen los demás acerca de él o ella?

	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.
Fuente	<ul style="list-style-type: none"> 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

Nombre de la práctica	Obtener lo que quieres.
Descripción de la práctica	Reflexionarás sobre alguna meta que desees alcanzar y propondrás una forma de conseguirla.
Palabras clave	Logro, involucramiento, fortalezas de carácter, esperanza, autorregulación, metas y objetivos a largo plazo.
Instrucciones para el aprendiz	<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. Para que puedas generar esta guía, responde las siguientes preguntas:</p> <ol style="list-style-type: none"> 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. 2. ¿Qué te impide que lo tengas en este momento? 3. ¿Qué sufrimiento estás experimentando en tu vida por no tenerlo en este momento? 4. ¿Qué placer, involucramiento, relación, significado o logro tendrías en tu vida si tuvieras eso en este momento? 5. ¿Qué hábitos te detienen o no te dejan avanzar hacia eso que quieres? 6. ¿Qué nuevos hábitos podrías generar para ayudarte a obtener lo que quieres? 7. ¿Qué dos cosas podrías hacer para romper con los hábitos que no te permiten avanzar hacia lo que quieres y generar hábitos nuevos? 8. ¿Te comprometes a hacer esas dos cosas? Si es así, ¿cuándo las harás? <p>Escribe tus resultados en un sitio donde puedas verlos constantemente.</p>
Fuente	<ul style="list-style-type: none"> 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

Nombre de la práctica	Felicidad en el trabajo.
Descripción de la práctica	Reflexionarás sobre las distintas dimensiones de tu vida cotidiana, enfocando el análisis a cómo fomentar un estado de ánimo y relaciones positivos en el ámbito laboral.

Palabras clave	Involucramiento, emociones positivas, relaciones positivas.
Instrucciones para el aprendizador	<p>Elegir conscientemente maneras de incrementar la felicidad en el trabajo puede hacer la diferencia en cómo nosotros 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 que están enfocadas en distintas dimensiones de tu vida:</p> <ul style="list-style-type: none"> • Dar: ¿cómo estoy apoyando a mis colaboradores, compañeros, líderes, proveedores y clientes? • Relaciones: ¿cómo puedo mejorar mis relaciones en el trabajo?, ¿cómo logro un balance entre la vida laboral y familiar? • Ejercicio: ¿cómo puedo integrar la actividad física dentro de mis actividades diarias?, ¿cómo aseguro que estoy comiendo bien y descansando lo suficiente? • Conciencia: ¿cómo puedo construir momentos de atención plena en mi día laboral? • Ensayo: ¿qué habilidades estoy construyendo?, ¿qué cosas nuevas he experimentado? • Dirección: ¿cuáles son mis metas laborales hoy, esta semana, este año?, ¿cómo caben y contribuyen estas con mis metas de vida y me ayudan a desarrollar mis competencias en la construcción de mis relaciones y cómo contribuyo con lo anterior a ayudar a otros?, ¿cómo se pueden alinear mis metas laborales con las de mi equipo y la organización? • Resiliencia: ¿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? • Emoción: ¿qué cosas, aunque sean pequeñas, puedo encontrar que me pueden hacer sentir bien en mi trabajo hoy?, ¿qué me ha hecho sonreír?
Fuente	Tomado del Catálogo de actividades para profesores.

Práctica 9

Nombre de la práctica	Interacciones positivas.
Descripción de la práctica	Reflexionarás sobre las cualidades positivas que aprecias de las personas con las que interactúas diariamente.

Palabras clave	Relaciones positivas.
Instrucciones para el aprendizador	<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 diariamente ayuda a incrementar nuestra 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 y/o colegas y piensa en las siguientes preguntas. En cada caso, anota ejemplos específicos.</p> <ol style="list-style-type: none"> 1. ¿Qué te atrajo de tus compañeros cuando se conocieron? 2. ¿Qué cosas han disfrutado al hacerlas juntos? 3. ¿Qué cosas realmente aprecias de ellos en este momento? 4. ¿Cuáles son sus fortalezas? <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"> • “Realmente me encanta cuando ellos...”. • “Son tan buenos para...”. • “Viéndolos hacer..., me recuerda ese fantástico día cuando nosotros...”. <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>
Fuente	Basado en el Catálogo de actividades para profesores.

Práctica 10

Nombre de la práctica	Las fortalezas se muestran en nuestras historias.
Descripción de la práctica	Reflexionarás sobre las fortalezas de carácter que aplicaste en una situación.
Palabras clave	Fortalezas de carácter.

Instrucciones para el aprendizador	<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>El siguiente enlace es externo a la Universidad Tecmilenio, al acceder a este considera que debes apegarte a sus términos y condiciones.</p> <p>http://www.viacharacter.org/www/Character-Strengths/VIA-Classification</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"> 1. Describe detalladamente, mediante 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. 2. Puede ser cualquier suceso que te haya marcado por la manera en que te desarrollaste. 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? 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. 5. Observa y clasifica cuáles son las fortalezas que usaste en tu anécdota. Reflexiona sobre el impacto que estas pueden tener en tu desempeño cotidiano.
Fuente	<p>Niemiec, R. (2016). <i>How to Assess Your Strengths: 5 Tactics for Self-Growth</i>. Recuperado de https://www.psychologytoday.com/us/blog/what-matters-most/201603/how-assess-your-strengths-5-tactics-self-growth</p>

Práctica 11

Nombre de la práctica	Tus fortalezas en los ojos del otro.
Descripción de la práctica	En la práctica podrás reflexionar sobre la percepción que otros tienen sobre tus fortalezas de carácter.
Palabras clave	Fortalezas de carácter.
Instrucciones para el aprendizador	¿Recuerdas alguna ocasión en la que hablaste con algún colega y este te reveló algo positivo que piensa de ti? Cuando esto ocurre, usualmente deja huella en nuestros comportamientos y acciones, pues nos damos cuenta de que las personas tienen percepciones sobre nuestras fortalezas que nosotros mismos no vislumbramos. Haz lo siguiente:

	<ol style="list-style-type: none"> 1. Piensa sobre alguna vez que algún compañero de trabajo te compartió lo que piensa de ti y que te haya sorprendido. 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? 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.
Fuente	<p>Niemiec, R. (2016). <i>How to Assess Your Strengths: 5 Tactics for Self-Growth</i>. Recuperado de https://www.psychologytoday.com/us/blog/what-matters-most/201603/how-assess-your-strengths-5-tactics-self-growth</p>

Práctica 12

Nombre de la práctica	Plantea tus objetivos como metas de aproximación y replantea tus metas de evitación.
Descripción de la práctica	Con base en lo que plantea Grenville (2012), en la práctica podrás definir diferentes tipos de metas y encontrar la mejor manera de conseguirlas.
Palabras clave	Objetivos, metas y planes.
Instrucciones para el aprendizador	<p>La autora Bridget Grenville-Cleave (2012) comenta que en el establecimiento de metas es importante distinguir los tipos de metas que hay y menciona dos:</p> <ol style="list-style-type: none"> 1. Metas de aproximación (<i>approach</i>): son las metas con resultados positivos (deseables, placenteros, benéficos o que nos gustaría tener) y hacia las cuales trabajamos. 2. Metas de evitación (<i>avoidance</i>): son las metas con resultados negativos (indeseables, dolorosos, dañinos, o nos disgustan) y en las cuales trabajamos para evitarlas. <p>Ejemplo:</p> <p>Meta de aproximación:</p> <ul style="list-style-type: none"> • Ser más eficiente. • Ser amigable y extrovertido en reuniones. • Asumir el rol de líder en el trabajo. <p>Meta de evitación:</p> <ul style="list-style-type: none"> • Dejar de aplazar. • Dejar de ser tan tímido en las reuniones. • No pasar desapercibido en el trabajo. <p>Las investigaciones que se han realizado respecto a estos tipos de metas muestran que perseguir metas de evitación resulta en un detrimento</p>

	<p>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">• ¿Qué tipo de metas te has planteado tú?• ¿Hay algunas metas que puedas replantear en una forma más positiva?• ¿Cuándo las tendrás listas?
Fuente	Grenville, B. (2012). <i>GOAL-SETTING SECRETS</i> . Recuperado de http://positivepsychologynews.com/news/bridget-grenville-cleave/2012013120696