



PLANIFICACIÓN ANUAL

Escuela:	PROGRAMA DE DESARROLLO EDUCATIVO - NIVEL SECUNDARIO - ESCUELA EXPERIMENTAL CON ÉNFASIS EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN. PROA LAS VARILLAS.
Curso:	5° Año
Ciclo:	Ciclo Orientado
Modalidad:	Común
Orientación:	Desarrollo en Software
Espacio curricular:	Programación III
Cursante:	Chiocarello, Lorenzo

Tutor/a:	

Fundamentación	<p>En el marco de la institución educativa PRoA Las Varillas, donde el enfoque pedagógico está orientado hacia la formación integral en tecnologías, la planificación de Programación III se dirige a los estudiantes de 5to año. Este grupo se caracteriza por su interés y capacidad en el ámbito de la informática, así como por su progreso en los conocimientos previos adquiridos en los años anteriores, lo que los coloca en una posición favorable para abordar temáticas avanzadas en programación.</p> <p>La dimensión político-pedagógica está vinculada a la planificación, organización y ejecución de propuestas educativas, las cuales buscan no solo gestionar las dinámicas en el aula, sino también adaptar el proceso de enseñanza a las necesidades contextuales y cognitivas de los estudiantes. Desde la perspectiva de Feldman (2010), la planificación de propuestas educativas tiene un papel central en la orientación de las prácticas docentes, permitiendo un uso adecuado del tiempo para diseñar estrategias que respondan a las demandas del aula. Meirieu (1998), por su parte, destaca que aunque exista una incertidumbre inherente en cada encuentro educativo, es necesario intentar controlar las condiciones del proceso de enseñanza, lo que subraya la importancia del rol del docente como mediador y organizador de los aprendizajes. Este enfoque resalta la importancia de un docente que no sólo transmita contenidos, sino que se posicione como un organizador del proceso de aprendizaje, capaz de ajustar y transformar su práctica según el contexto político y pedagógico. Siguiendo este enfoque se puede enriquecer las clases tomando en cuenta una</p>
----------------	--

planificación estratégica y flexible, adaptándose a las necesidades del contexto, donde tener estrategias claras que permitan abordar las singularidades de cada grupo de estudiantes y que se ajusten a actividades según sus ritmos de aprendizaje y contextos. A su vez se debe organizar el contenido de manera secuencial estructurado en clases en secuencias didácticas que permitan a los estudiantes profundizar en los temas.

La planificación educativa, como señala Feldman (2010), es esencial no solo para controlar lo que ocurre en el aula, sino para orientar las prácticas docentes y diseñar estrategias adecuadas. Esto incluye la selección, organización y secuenciación de contenidos. Esta tarea no es estática; por el contrario, implica ajustes continuos durante todo el proceso de enseñanza, ya que los contenidos que se seleccionan para enseñar no son neutros, sino que implican decisiones de inclusión y exclusión (Terigi, 2020). En la planificación, también es importante tener en cuenta el tiempo que requiere el aprendizaje. La organización de la enseñanza debe permitir abordar los temas en profundidad mediante secuencias didácticas o proyectos, aunque no todos los temas pueden ser tratados de este modo. En este sentido, Litwin (2014) resalta la importancia de las "ayudas para aprender", que facilitan la comprensión de los temas al vincularlos con conocimientos previos y al elaborar conclusiones que favorezcan la abstracción. Elaborar una propuesta de enseñanza implica seleccionar contenidos, establecer vínculos y definir estrategias que acerquen a los estudiantes a los temas propuestos. Edelstein (1996) destaca la necesidad de que estas estrategias surjan de una construcción metodológica particular, considerando el contexto y los desafíos cognitivos de los estudiantes. Para hacer significativa la enseñanza, es recomendable partir de situaciones problemáticas o temas actuales, generando interés y permitiendo la reflexión (Sadovsky, 2020). Además, las actividades deben ser variadas pero no excesivamente complejas, y las consignas claras y bien formuladas (Litwin, 2014).

	<p>Al seleccionar contenidos, se tiene que tener consideración no sólo qué es relevante para la disciplina, sino también qué es significativo y accesible para mis estudiantes. Además, es importante organizar esos contenidos de manera secuencial, permitiendo una comprensión progresiva y respetando el tiempo necesario para aprender cada tema en profundidad. Utilizaré materiales variados y adaptados a las necesidades y características de mi alumnado, integrando recursos digitales cuando sea adecuado y asegurándose de que sean comprensibles y útiles para facilitar el aprendizaje. Los contenidos son seleccionados a partir del diseño curricular pero adaptándose a las necesidades y conjuntos de habilidades que poseen y se desea alcanzar.</p> <p>En cuanto a las capacidades fundamentales que se favorecen, la propuesta fomenta el pensamiento crítico, el trabajo en equipo, la autonomía en la resolución de problemas, y el manejo de herramientas tecnológicas actuales. Estas capacidades se desarrollan a través de estrategias como el trabajo por proyectos, la aplicación práctica de los conceptos y la integración de tecnologías relevantes. Estas elecciones forman parte de un acuerdo institucional que prioriza el desarrollo integral de los estudiantes, preparándose no solo para el ámbito académico, sino también para un futuro profesional en un mundo digitalizado. El enfoque es desarrollar capacidades y lograr objetivos de aprendizaje a través del contenido curricular, seleccionando situaciones o problemas que permitan aplicar dichos contenidos de forma significativa y relevante para los estudiantes.(Escuela Posible, M. de Educación de Córdoba).</p>
Objetivos	<ul style="list-style-type: none">- Desarrollar el backend con Node.js y Express, crear servidores, gestionar rutas y conectarse a base de datos.- Construir interfaces con React, crear componentes, manejar estados y navegar entre vistas.- Integrar el backend con el frontend y consumir los datos utilizando Fetch API y Axios.

Contenidos y aprendizajes	Backend con JavaScript <ul style="list-style-type: none">- Fundamentación de Node.js y su utilización.- Creación de servidores, manejo de rutas y controladores con Express.- Desarrollo de proyecto Backend con gestión de base de datos MySQL. Frontend con JavaScript <ul style="list-style-type: none">- Fundamentación, creación de componentes y funcionalidad de React.- Manejo de estados, efectos y rutas.- Conexión con Fetch API y Axios para consumir el backend desde el frontend.
Metodología de enseñanza	<p>La metodología de enseñanza se basa en un enfoque constructivista y práctico, que integra conocimientos teóricos con proyectos reales. El enfoque progresivo y basado en proyectos busca desarrollar habilidades clave como la resolución de problemas complejos, autonomía en el aprendizaje, y capacidad de colaboración. A través de ejercicios prácticos y proyectos integradores, los estudiantes adquirirán competencias fundamentales para enfrentar desafíos en entornos reales, fomentando su capacidad de análisis, adaptación y trabajo en equipo.</p> <p>La teoría será presentada en cuadernillos que contienen conceptos y ejemplos esenciales. El docente explicará estos contenidos y acompañará a los estudiantes para asegurar su comprensión. Posteriormente, los estudiantes pondrán en práctica los ejemplos, adaptándolos y aplicándolos en el proyecto que desarrollarán durante todo el ciclo. Esta metodología se basa en el aprendizaje experiencial, donde el "aprender haciendo" cobra protagonismo, permitiendo a los estudiantes adquirir conocimientos a través de la experiencia directa. Este enfoque favorece el desarrollo de habilidades como la resolución de problemas, el pensamiento crítico y el trabajo colaborativo, promoviendo una mayor autonomía en el abordaje de situaciones problemáticas.</p>

Cronograma tentativo	<p>2 clases semanales de 80 minutos cada una.</p> <ul style="list-style-type: none">- Diagnóstico: 4 semanas.- Eje N1: 14 semanas.- Eje N2: 16 semanas
Evaluación	<p>Se realizarán 2 instancias evaluativas con la presentación del proyecto finalizado en cada eje.</p> <p>Los estudiantes crearán su portafolio donde cada clase irán subiendo los avances y los diferentes ejercicios que deberán realizar. La evaluación será de forma formativa en esta instancia porque habrá una retroalimentación constante.</p> <p>Los instrumentos a evaluar serán el proyecto finalizado en cada eje, con su presentación y explicación de los conceptos adquiridos.</p> <p><u>Criterios de evaluación:</u></p> <ul style="list-style-type: none">- Criterios generales:<ul style="list-style-type: none">- Claridad en la organización del código: El estudiante debe demostrar que organiza su código de manera modular y comprensible, tanto en el backend como en el frontend.- Cumplimiento de buenas prácticas: Se evaluará si el estudiante sigue buenas prácticas de desarrollo, como el manejo adecuado de errores, el uso de nombres de variables claros y el mantenimiento de un código limpio y comentado.- Capacidad de resolución de problemas: El estudiante debe demostrar la capacidad de identificar y solucionar problemas en su código, tanto en el backend como en el frontend, utilizando herramientas de depuración cuando sea necesario.

- Integración correcta entre frontend y backend: Se evaluará si el estudiante puede integrar correctamente ambos lados de la aplicación, asegurando que las peticiones del frontend son recibidas y procesadas adecuadamente en el backend.

Eje N1:

- Identifica qué es Node.js, cuándo y porqué se utiliza.
- Realiza una configuración básica de un entorno de desarrollo, utilizando las herramientas de npm para el manejo de paquetes y módulos.
- Demuestra cómo importar y exportar módulos y como es la organización de un proyecto en Node.js, identifica componentes, carpetas y archivos.
- Crea y configura un servidor con Express, definiendo puertos, middleware y manejo de errores.
- Configura rutas en Express, tanto rutas básicas (GET, POST, PUT, DELETE) como rutas dinámicas, y debe ser capaz de explicar cómo estas rutas permiten la interacción entre cliente y servidor.
- Implementa controladores de forma modular, entendiendo cómo separar la lógica de negocio de las rutas para mantener un código limpio y escalable.
- Capacidad de usar middleware en Express para gestionar tareas como la validación de datos, autenticación o manejo de errores.
- Demuestra cómo conectar un servidor Node.js/Express con una base de datos MySQL utilizando bibliotecas como mysql2 o sequelize. Debe configurar correctamente las credenciales, puertos y parámetros de conexión.
- Identifica, crea tablas y define relaciones utilizando sentencias SQL.
- Realiza consultas básicas de CRUD en MySQL conectadas con el proyecto.

	<p>Eje N2:</p> <ul style="list-style-type: none">- Identifica y explica fundamentos de React, como el Virtual DOM, los componentes, la jerarquía de componentes, y el ciclo de vida de los componentes.- Demuestra la capacidad de crear y reutilizar componentes funcionales y de clase en React. Debe ser capaz de organizar componentes de manera eficiente y modular.- Demuestra cómo utilizar props para pasar información entre componentes y cómo gestionar el estado interno de un componente con useState y setState. Debe ser capaz de manejar cambios en el estado y cómo estos afectan el renderizado.- Implementa y explica un sistema de rutas dentro de una aplicación React utilizando React Router, configurando rutas dinámicas, navegación entre componentes y redirecciones.- Capacidad de realizar peticiones HTTP (GET, POST, PUT, DELETE) utilizando Fetch API, manejar respuestas y errores, y procesar datos obtenidos del backend.- Demuestra el uso de Axios para realizar las mismas operaciones que Fetch, explicando las ventajas o diferencias entre ambos.- Capacidad de manejar peticiones asíncronas y entender el concepto de promesas, async/await, y cómo manejar los estados de carga y errores en la interfaz de usuario al consumir APIs.
Bibliografía	Referenciar los materiales obligatorios y complementarios que se ofrecerán en el espacio curricular. Pueden ser textos, materiales audiovisuales, fichas de cátedra, documentos provinciales o nacionales, entre otros. Al momento de consignarlos, se deben seguir los lineamientos de las normas APA.

- | | |
|--|--|
| | <ul style="list-style-type: none">- Documentación oficial de Node.js (2024).
https://nodejs.org/en- Documentación oficial de React (2024).
https://es.react.dev/- Front End Development Libraries. freeCodeCamp (2024).
https://www.freecodecamp.org/learn/front-end-development-libraries/- Documentación oficial de Axios (2024).
https://axios-http.com/es/docs/intro- W3Schools Online Web Tutorials - MySQL (2024)
https://www.w3schools.com/mysql/default.asp- Producciones (textos y presentaciones) del profesor basadas en la bibliografía citada y otros contenidos. |
|--|--|

Secuencia didáctica- Residencia Profesional Docente

Modalidad	ProA
Espacio curricular	Programación III
Tema	Eje N1: Backend con JavaScript Contenido: Creación de servidores, manejo de rutas y controladores con Express. Tema: Implementación y creación de controladores y rutas
Curso	5to año
Tutor	Chiocarello, Lorenzo
Coformador	-

Contextualización de la propuesta.	Sintetice las características del grupo clase que fueron indagadas en la Instancia 1 que justifiquen las decisiones tomadas haciendo foco en la heterogeneidad del aula y los diversos modos en los que los estudiantes se relacionan con el contenido. (200 palabras aprox.)
------------------------------------	---

	<p>En el marco de la educación tecnológica integral del PProA Las Varillas, la planificación de la secuencia didáctica está diseñada para un aula heterogénea compuesta por 17 estudiantes de 5to año. La diversidad del grupo se manifiesta en sus trayectorias académicas, estilos de aprendizaje y niveles de autonomía, lo que requiere una estrategia pedagógica flexible y adaptativa.</p> <p>En este sentido, se integrarán actividades teóricas y prácticas, priorizando un enfoque basado en proyectos (ABP) que permita a los estudiantes aplicar los contenidos en situaciones significativas y concretas, donde deban llevar los conocimientos y soluciones a situaciones reales, acompañando la teoría y la práctica.</p> <p>Durante las clases observadas, se advirtió que si bien la mayoría mantiene la atención durante las explicaciones orales apoyadas en proyecciones visuales, algunos presentan dificultades para sostener el foco o consultar por iniciativa propia. Por ello, se proponen actividades que equilibren la autonomía con instancias de acompañamiento docente, habilitando distintos ritmos de trabajo. Las explicaciones claras, los ejemplos proyectados y los ejercicios individuales en computadora son recursos clave en este proceso.</p> <p>A su vez, se buscará fomentar instancias de autoevaluación y reflexión para que los estudiantes puedan reconocer sus avances, dificultades y estrategias personales. Aunque la interacción entre pares no es espontánea en todos los casos, se plantea el trabajo colaborativo como una oportunidad para promover el intercambio, potenciar el aprendizaje entre compañeros y fortalecer habilidades sociales y comunicativas.</p> <p>De este modo, la propuesta no solo apunta a la apropiación de saberes técnicos, sino también al desarrollo de competencias vinculadas al pensamiento crítico, la resolución de problemas y la preparación para un entorno digital dinámico, teniendo en cuenta la dimensión socioemocional, la realidad particular y los intereses personales de cada estudiante.</p>
--	--

	Propuesta
Fundamentación	<p>Expresar una fundamentación sobre la selección y la jerarquización de los contenidos de la secuencia didáctica.</p> <p>Explicitar el porqué de ese recorte y organización del contenido para este curso en particular.</p> <p>Esta producción deberá dar cuenta de la articulación dialógica entre contenidos, objetivos y tiempo asignado para su desarrollo.</p> <p>La extensión deberá ser de entre 300 y 500 palabras.</p> <p>Para resolver este punto, pueden recuperar parte del contenido elaborado durante el Momento 1</p> <p>La selección y jerarquización de los contenidos en esta secuencia didáctica responde a criterios pedagógicos que buscan garantizar un aprendizaje significativo y aplicable en el tiempo disponible. Se han priorizado los temas esenciales del desarrollo backend con Flask, enfocándose en la implementación de controladores y rutas debido a su centralidad en la arquitectura de aplicaciones web modernas.</p> <p>Como señala Terigi (2020), la organización de los contenidos debe permitir una apropiación efectiva, vinculando nuevos conocimientos con saberes previos. En este caso, los estudiantes ya poseen bases en programación en Python, por lo que la secuencia se estructura de manera progresiva, partiendo de conceptos fundamentales de Flask y avanzando hacia la modularización y buenas prácticas en el desarrollo backend. Esta progresión facilita la comprensión y aplicación de los contenidos en proyectos concretos, reforzando la autonomía y la capacidad de resolución de problemas.</p> <p>El recorte de contenidos responde a la necesidad de equilibrar la profundidad del aprendizaje con el tiempo disponible. La</p>

	<p>propuesta se desarrolla en tres clases, donde se abordan de manera articulada los objetivos de aprendizaje: comprender la estructura de controladores en Flask, manejar rutas y solicitudes HTTP, y aplicar estos conceptos en una aplicación funcional. Este enfoque permite integrar conocimientos teóricos con prácticas significativas, alineando la secuencia con un enfoque activo y participativo (Litwin, 2014).</p> <p>Asimismo, se promueve un aprendizaje basado en desafíos reales, incentivando el pensamiento crítico y la toma de decisiones (Sadovsky, 2020). Se utilizan estrategias didácticas flexibles, combinando exposiciones teóricas con resolución de problemas y trabajo en equipo, facilitando así una construcción de conocimiento adaptada a los ritmos de aprendizaje del grupo.</p> <p>Esta secuencia didáctica no solo busca desarrollar habilidades técnicas en Flask, sino también fortalecer competencias transversales como la organización del código, la colaboración y la capacidad de enfrentar problemáticas reales del desarrollo web. A través de una planificación estratégica y progresiva, se garantiza que los estudiantes logren apropiarse de los contenidos en un período breve.</p>
Metodología	<p>Expresar las decisiones y las diversas estrategias en relación con el abordaje de los contenidos y las capacidades que se pretenden desarrollar.</p> <p>Aprendizaje Basado en Problemas (ABP): Esta metodología favorece un pensamiento estratégico, permitiendo a los estudiantes enfrentar problemas reales relacionados con la programación y proponer soluciones innovadoras. Se busca incentivar el desarrollo del pensamiento crítico y la creatividad.</p> <ul style="list-style-type: none">● Contenidos: Se trabajarán situaciones problemáticas relacionadas con el uso y desarrollo de software, donde los estudiantes deberán analizar, proponer soluciones y aplicar sus conocimientos en programación.

	<ul style="list-style-type: none"> ● Intervenciones Docentes: El docente facilitará la identificación de problemas, guiará la viabilidad de las posibles soluciones y fomentará la toma de decisiones informadas. ● Formato Pedagógico: Talleres donde los estudiantes podrán experimentar con diversas estrategias de resolución de problemas. <p>Aula-Taller: Se promoverá un enfoque de "aprender haciendo", donde la práctica y la producción serán fundamentales para la adquisición de conocimientos.</p> <ul style="list-style-type: none"> ● Contenidos: Se implementarán actividades prácticas en las que los estudiantes manipularán herramientas y entornos de desarrollo, aplicando conocimientos adquiridos en clases teóricas. ● Intervenciones Docentes: Se incentivará la experimentación, la interacción grupal y la producción de conocimiento a través del desarrollo de proyectos. ● Formato Pedagógico: Espacios de trabajo colaborativo donde se combine la práctica con la reflexión teórica.
Formato curricular	<p>Seleccionar alguno de los siguientes formatos para el desarrollo de la unidad didáctica: materia/asignatura; taller; ateneo; seminario; laboratorio; observatorio; proyecto o trabajo de campo. Justifiquen esa elección</p> <p>Para el desarrollo de esta unidad didáctica, se ha seleccionado el formato de taller. Este enfoque permite un aprendizaje activo y participativo, en el que los estudiantes pueden aplicar conocimientos teóricos en un contexto práctico y significativo.</p> <p>La elección del taller se fundamenta en los siguientes aspectos:</p>

	<ul style="list-style-type: none"> • Aprendizaje basado en la práctica: Los estudiantes trabajan con problemas reales de programación, promoviendo el desarrollo de habilidades técnicas y de pensamiento crítico. • Interacción colaborativa: Se fomenta el trabajo en equipo y el intercambio de ideas, enriqueciendo el proceso de aprendizaje. • Flexibilidad y adaptabilidad: Permite ajustar las actividades a las necesidades individuales y grupales, favoreciendo una enseñanza personalizada. • Enfoque integral: Combina la teoría con la experimentación, garantizando una comprensión más profunda y significativa de los contenidos abordados.
Aprendizajes y contenidos	<p>Explicite en esta sección los aprendizajes y contenidos. Considerar para ello el Diseño Curricular de la modalidad y sus actualizaciones.</p> <p>Backend con JavaScript</p> <ul style="list-style-type: none"> - Fundamentación de servidores en Node.js y Express. - Implementación de rutas para manejar peticiones HTTP. - Creación y utilización de controladores para manejar la lógica del servidor. - Manejo de respuestas JSON y códigos de estado HTTP.
Propósitos de enseñanza	<p>Explicite qué se pretende enseñar, esto es, las finalidades formativas que organizan su propuesta de enseñanza. Tenga en cuenta que las acciones que se consignan aquí corresponden al docente.</p>

	<ul style="list-style-type: none">- Promover la comprensión y aplicación de los fundamentos del desarrollo backend con JavaScript, Node.js y Express como herramientas esenciales en el desarrollo de aplicaciones web modernas.- Facilitar la adquisición de habilidades técnicas para la implementación de servidores, rutas y controladores, fomentando buenas prácticas de programación y organización del código.- Generar espacios de práctica significativa donde los estudiantes puedan experimentar con problemas reales del desarrollo web, estimulando su autonomía y capacidad para la resolución de desafíos técnicos.- Propiciar el desarrollo del pensamiento lógico-computacional mediante la implementación de soluciones a problemas de programación backend, articulando conocimientos previos con los nuevos aprendizajes.- Favorecer el trabajo colaborativo como estrategia para la construcción colectiva del conocimiento y el intercambio de saberes entre pares.- Acompañar a los estudiantes en su proceso de aprendizaje, atendiendo a la diversidad de ritmos y estilos mediante intervenciones personalizadas y recursos didácticos variados.
Objetivos Considerar el material “Apuntes para la diversificación”	<p>Explicite qué se pretende que los estudiantes aprendan durante el desarrollo de esta secuencia didáctica. Puede responder a la pregunta: ¿qué esperamos que nuestros estudiantes logren? Tengan en cuenta que estos deberán ser alcanzables a partir de la propuesta didáctica.</p> <p>Contemple si es posible, deseable o necesario diversificar objetivos de aprendizaje.</p> <ul style="list-style-type: none">- Implementar un servidor básico utilizando Node.js y Express, configurando adecuadamente los parámetros necesarios para su funcionamiento.- Diseñar y estructurar rutas de API REST para gestionar diferentes tipos de peticiones HTTP (GET, POST, PUT,

	<p>DELETE) de manera organizada y eficiente.</p> <ul style="list-style-type: none"> - Desarrollar controladores modulares que separen la lógica del negocio de las rutas con el manejo de la base de datos.
<p>Demandas cognitivas</p> <p>Considerar el material “Apuntes para la diversificación” y la Clase 4 de Bases Didácticas.</p>	<p>Explicite las demandas cognitivas que se movilizan en relación con los objetivos propuestos. Algunos ejemplos: reconocer, describir, comparar, explicar, hipotetizar.</p> <p>Contemple si es posible, deseable o necesario proponer diverso grado de intensidad en las demandas cognitivas (para atender a distintos grados de complejidad de relación con el saber).</p> <ul style="list-style-type: none"> - Reconocer los elementos fundamentales de un servidor y comprender su importancia en el desarrollo backend con Node.js y Express. - Describir el funcionamiento de las rutas y los controladores en Express, además de la gestión de las peticiones HTTP. - Aplicar los conocimientos adquiridos para desarrollar una API REST básica con Node.js y Express, configurando correctamente sus rutas y controladores.
<p>Evaluación</p> <p>Considerar: “Apuntes para la diversificación” y Clase 6, “Repensar la evaluación en el aula”</p>	<p>Defina instrumentos de seguimiento y evaluación en el marco de la propuesta. Redacte criterios de evaluación considerando los objetivos de aprendizaje propuestos.</p> <p>Considere las decisiones previas en torno a la diversificación para definir en qué caso es necesario diversificar la evaluación.</p> <p><u>Instrumentos:</u></p>

(Compilación de Clases de Práctica)	<ul style="list-style-type: none"> - Rúbrica <p><u>Evaluación:</u></p> <ul style="list-style-type: none"> - Formativa y procesual <p><u>Criterios de evaluación:</u></p> <ul style="list-style-type: none"> - Implementa y comprende la creación de servidores con Node.js y Express: Se evaluará la capacidad del estudiante para configurar un servidor básico con Express, asegurando su correcto funcionamiento. Se observará si el estudiante comprende la estructura y propósito de un servidor en el desarrollo backend. - Diseña y gestiona rutas en una API REST: Se evaluará si el estudiante implementa correctamente rutas para manejar peticiones HTTP. - Desarrolla controladores para la lógica del servidor: Se analizará si el estudiante implementa correctamente los controladores modulares que separen la lógica del negocio de las rutas.
-------------------------------------	--

Boceto/ esquema de las clases: Describimos sintéticamente ¿qué queremos que ocurra en cada clase y por qué?

Clase 1:

Se busca que los estudiantes comprendan el fundamento de servidor en Node.js y cómo Express facilita la gestión de rutas. Se busca que configuren su primer servidor y pueda ejecutarse correctamente. Es fundamental que entiendan la base sobre la que trabajarán las clases siguientes y puedan ver un resultado inmediato que motive.

Clase 2:

Se busca que los estudiantes comprendan la lógica de una API REST y puedan implementar rutas que respondan a distintos tipos de

solicitudes. Cada solicitud mostrará un mensaje que luego será modificado con las diferentes respuestas del controlador. Esto permitirá que puedan dominar las rutas permitiéndoles organizar la lógica de los proyectos y que puedan conectarse con bases de datos o frontends en el futuro.

Clase 3:

En esta clase se introducirá el concepto de controladores, separando la lógica de las rutas en funciones externas. Cada ruta utilizará ahora una función ubicada en un archivo distinto. Se espera que cada estudiante pueda implementar su propio controlador(archivo) y comprender su funcionamiento.

Guion conjetural - Clase por clase

Escriban cómo imaginan que será cada clase momento a momento: qué dirán, qué harán, qué propondrán hacer. Incluyan referencias a los propósitos de enseñanza, los objetivos de aprendizaje, las estrategias metodológicas, las actividades y recursos. Justifiquen las decisiones tomadas. En cada caso, consideren si es posible, deseable o necesario planear alternativas para las actividades o recursos trabajados.

Es importante que el texto sea claro, no sólo porque orientará sus acciones futuras, sino porque sus colegas y tutor lo leerán para poder hacerle aportes y deben ser capaces de imaginar cómo se desarrollarán sus clases de práctica a partir de una simple lectura.

El guión conjetural es dinámico, por lo que pueden ir incorporando cambios a medida que reciben sugerencias. Sin embargo, les pedimos que no resuelvan los comentarios al interior del documento para que sea posible “rastrear” las conversaciones que dieron lugar a la producción final.

Clase 1:

En el inicio de la clase, se propondrá que objetivos y a qué punto se desea llegar en esta clase, a la configuración de un servidor. Será importante destacar con una pregunta orientadora sobre qué pasa realmente, cuando escribis una dirección de un sitio web y le das Enter. Los estudiantes reflexionarán, comentarán y discutirán entre ellos sobre qué es lo que pasa. Aquí será importante encaminarlos hacia el concepto de “servidor: es quien responde los pedidos”.

Se proyectará un servidor corriendo localmente para hacerle una petición y que me muestre un mensaje. Se introducirán además los conceptos de Node.js y Express que uno es un marco de trabajo para escribir un lenguaje del lado del backend y que el otro es un framework que nos simplificará cosas.

Harán un registro de los conceptos teóricos para después interiorizarlos.

A continuación, los estudiantes formarán grupos de trabajo según estén sentados en las mesas, promoviendo el trabajo en equipo. Se les indicará paso a paso, escribiendo en el pizarrón y con la ayuda del proyector como deberán crear un proyecto de Node.js y como instalar Express. En este momento, los estudiantes irán copiando el código y tratando de no cometer errores ortográficos. Si en algún caso, algún grupo resulta que tiene errores, se les invitara a que con su compañero puedan solucionar dicho problema. Luego tendrán que probar que dicho servidor funciona y generará una sensación de logro rápido, al ver que “funciona”.

Una vez que los grupos logren que su servidor funcione, se los invita a experimentar cambiando el texto del mensaje o el número de puerto para explorar mínimamente los elementos que configuran el servidor.

Habrà un momento al final, donde los estudiantes que no pudieron resolver los problemas puedan consultar con otros compañeros o con el profesor para arreglar dicho problema.

Clase 2:

En esta clase se profundiza sobre el uso de las rutas en Express, el docente iniciara con una pregunta orientadora sobre si se acuerdan que pasaba cuando levantamos el servidor y entramos al localhost para ver nuestra página, y que pasaria si queríamos mostrar otra cosa o que nuestra URL sea diferente. A modo de lluvias de ideas se van a ir anotando en el pizarrón y los estudiantes propondrán cambiar dichas rutas según el proyecto o propósito que quieren darle una solución. De este modo se proyectara una ruta de tipo POST que recibe un parametro y como cambia la URL cuando entro al navegador. Los estudiantes ahora deberan ir probando diferentes rutas y que devuelve o que muestra cada una. Se entusiasman al ver que responde según lo que escriben. Luego se propondrá una actividad guiada: crear una ruta `/doble/<numero>` que devuelva el doble del número recibido. Se va a circular en el aula mientras los grupos trabajan y hacen preguntas para

guiar el razonamiento. Para finalizar la clase, se abrirá un debate en el que los estudiantes responderán con sus palabras y darán soluciones a una pregunta ¿qué diferencia hubo entre esto y lo que hacíamos en la primera clase? Donde los guiará hacia que creando diferentes rutas pasan cosas diferentes.

CLASE 3:

En esta clase introduciremos el concepto de controlador, se preguntará a los estudiantes si recuerdan los conceptos de servidor y rutas para poder seguir con controladores.

Con una pregunta orientadora: Qué ventajas creen que tiene separar la lógica en funciones distintas, en lugar de poner todo en el mismo archivo? Se hará una lluvia de ideas y se anotarán en el pizarrón, guiando siempre las respuestas hacia orden, mantenimiento y reutilización. Se proyectará el código de un controlador y los estudiantes deberán crear una carpeta controller y definir las rutas antes creadas y crear un nuevo archivo controller para manejar esas rutas y peticiones. Aquí los estudiantes les tomará un poco más de tiempo realizar la actividad, consultaran con el docente y con sus compañeros.

El docente abrirá Postman en el proyector, hará una petición GET y mostrará cómo llega una respuesta. Los estudiantes sorprendidos por la nueva herramienta, la que tendrán en su computadora, deberán abrirla, ejecutar el servidor y probar las diferentes rutas que crearon y verán como lo que definieron en el controlador se mostrará en el programa. Ahora los estudiantes resolvieron un problema de un contexto real, en el que programaron una aplicación, que sigue estándares y que permite la escalabilidad del trabajo. Se realizará una breve puesta en común, atendiendo a todos los conceptos vistos hasta la clase, los estudiantes responderán en su mayoría con sus propias palabras sobre los conceptos y cómo los usaron para solucionar una problemática real. Se los invitara a pensar para las clases siguientes, ¿qué otras funcionalidades podría tener un controlador además de devolver un mensaje?

DIARIO DEL PROFESOR

Registro de clases

Institución: Escuela de gestión provincial

Curso: 5° año

Disciplina: Programación III

Fecha: 06/05/2025

Acerca de la institución

La institución está ubicada en un barrio ya consolidado desde hace años. Su presencia ha generado un movimiento inédito en la zona, donde la vida barrial no estaba acostumbrada a la circulación constante de jóvenes, docentes y actividades educativas cotidianas. Este cambio implicó, al principio, un proceso de adaptación mutua, pero con el tiempo se consolidó un vínculo de colaboración y pertenencia entre la escuela y su comunidad.

La escuela se caracteriza por ser abierta a la comunidad y por participar activamente en proyectos colectivos, encuentros interdisciplinarios y propuestas sociocomunitarias. Se desarrollan actividades integradoras como ferias, exposiciones y trabajos con organizaciones locales, lo que fortalece el lazo entre la formación académica y las realidades del entorno. En este sentido, la escuela se propone no solo formar estudiantes con competencias técnicas, sino también ciudadanos activos capaces de intervenir críticamente en su contexto.

En cuanto a la infraestructura, el edificio cuenta con cuatro aulas convencionales, destinadas al desarrollo de clases teóricas o grupales. A esto se suman tres laboratorios especializados, entre ellos un laboratorio de informática, que constituye el principal espacio para el dictado de clases relacionadas con programación y tecnologías digitales. Este laboratorio está equipado con computadoras individuales, conexión Wi-Fi de buena calidad, proyector y recursos que favorecen el trabajo colaborativo y autónomo de los estudiantes.

Todo esto da cuenta de una institución que apuesta por la innovación pedagógica, que valora el trabajo por proyectos, el uso intensivo de las TIC y que concibe el aprendizaje como un proceso colectivo, situado y significativo.

Acerca del tema

Eje: Backend con JavaScript

Contenidos y aprendizajes: Creación de servidores, manejo de rutas y controladores con Express.

Los estudiantes, organizados en grupos previamente conformados, elaboraron un informe sobre la aplicación que desarrollarán para dar respuesta a una problemática identificada. En esta clase, el objetivo fue que, utilizando un framework, pudieran crear y levantar un servidor, especificando el puerto y descargando las librerías necesarias para su correcto funcionamiento. La tarea consistió en que, con el acompañamiento del docente y el código proporcionado, cada grupo pudiera crear su propio servidor y mostrar un mensaje básico como resultado de una configuración inicial.

Con el informe que habían desarrollado las clases anteriores y exponiendo oralmente a sus compañeros pudieron dar evidencia y estudiar acerca de los diferentes conceptos como lenguaje de programación, framework, rutas y controladores. A su vez con el informe debían ya traer ideas sobre cómo se iban a llamar las rutas y sus controladores.

Desarrollo de la clase

Desde el ingreso de los estudiantes al espacio de informática a las 12:40, se percibía un ambiente cargado de expectativas. Al entrar los estudiantes al espacio, donde ya me ubicaba antes por tener clases con otro curso, varios estudiantes me saludaron con entusiasmo, y un estudiante me preguntó: “Hoy empezamos con la página, ¿no profe?”. Esa espontaneidad inicial ya hablaba de un interés genuino por lo que íbamos a abordar: la creación del primer servidor web utilizando Express.js. La clase había sido anunciada la semana anterior, cuando los estudiantes, organizados en grupos de acuerdo a interés y afinidades, habían tenido que elaborar un informe. Muchos llegaban con la idea de dar ese “primer paso real” hacia su proyecto, donde se proponía resolver tanto una problemática real como imaginaria dentro de un entorno web.

Para activar conocimientos previos, propuse una lluvia de ideas en el pizarrón con la consigna: “¿Qué entienden por servidor?”. Las respuestas variaron desde definiciones intuitivas —“que conecta algo”— hasta formulaciones más técnicas —“un servicio”. A medida que anotaba los aportes, los fui agrupando en ejes como host y base de datos. Esta instancia permitió no solo reconocer saberes previos, sino también planificar y secuenciar las micro intervenciones con los objetivos propuestos.

Antes de pasar al código, tomé un momento, pedí que hicieran silencio y apartaran la vista de la computadora y que me prestaran atención. Esto tomó su tiempo, aproximadamente 3 minutos para que los estudiantes se enfocaran en mi exposición, todo esto para establecer los objetivos de la clase y su conexión con el proyecto anual. Les recordé que Express nos iba a permitir construir el backend de sus aplicaciones en Node.js, y que todo lo que hiciéramos hoy sería la base de futuras clases. Este encuadre fue importante para dar

sentido al esfuerzo, sobre todo en un grupo tan heterogéneo en cuanto a ritmo de aprendizaje.

Con el grupo ya activado, comenzamos el primer bloque práctico: la instalación de Express y la creación del archivo `server.js`. Mi computadora ya estaba conectada al televisor del aula, desde donde proyecté los pasos y comandos necesarios que los estudiantes debían ejecutar en la terminal. Dos grupos necesitaron ayuda para acceder a la terminal o al archivo del proyecto; me acerqué a sus equipos y resolvimos juntos. Varios estudiantes se levantaron a preguntarme si podían sacar fotos del código; aunque les servía para guardar la información, esto hizo que se atrasaran y cometieran errores por copiar apurados.

En esos casos, opté por una intervención mínima: propuse que buscaran y corrigieran los errores por sí mismos, fomentando la autonomía y la autoverificación, en lugar de ofrecer soluciones inmediatas. A medida que pasaba el tiempo, les fui consultando si habían podido solucionar dichos problemas donde me comentaron que con ayuda de la inteligencia artificial ChatGPT habían podido solucionarlos, donde se evidencio que le pasaron el código del error a la IA. Esta estrategia, aunque fue más lenta resultó efectiva.

La clase se estructuró en mini objetivos claros: instalar Node.js, iniciar el proyecto con `npm init`, instalar Express, importar la librería, crear una aplicación con `express()`, definir una ruta básica con `app.get()` y levantar el servidor local con `app.listen()`. Esta secuencia fragmentada fue esencial para mantener la atención y evitar la sobrecarga de información. En el camino, surgieron dudas puntuales: errores en la escritura de comandos, comprensión del middleware, o por qué era necesario indicar el puerto en `app.listen()`.

Aunque la clase se guió por estos pasos, el código proyectado y la explicación oral fueron claves para que el grupo avanzara. Un estudiante preguntó si era obligatorio que el archivo se llamara `server.js`, lo que dio lugar a una breve conversación sobre convenciones, puntos de entrada definidos en `package.json` y buenas prácticas. Esto permitió introducir nociones sobre la estructura de un proyecto en Node sin alejarnos demasiado del objetivo inicial. En este sentido, surgieron dudas que no estaban previstas y tuvimos que irnos a conceptos más generales como la modularización del código, y cómo se organiza una aplicación en Node de manera escalable, donde se explico que no siempre se debe llamar asi pero las buenas practicas si lo indican, quedando su duda resulta, ya que se le consulto si había comprendido.

Un grupo no lograba ejecutar correctamente por errores de sintaxis. En ese momento, propuse que se ayudaran entre grupos, y uno de los estudiantes que había expuesto previamente sobre Node.js se acercó espontáneamente. Les ofreció ayuda, revisó el código con ellos y les explicó con paciencia dónde estaba el error. Esta intervención no solo resolvió el problema puntual, sino que también generó un valioso momento de aprendizaje entre pares.

A nivel actitudinal, la motivación se mantuvo alta durante toda la clase. Incluso quienes no intervenían oralmente seguían el desarrollo con atención y trabajaban de forma sostenida en sus computadoras. Cuando llegó el momento de ejecutar el servidor y apareció en la terminal el mensaje "Servidor escuchando en el puerto 3000", se generó una especie de mini celebración colectiva, donde se escuchaban exclamaciones como: "¡Vamos!", "¡Ahí está!". Un estudiante dijo en voz alta y para todo el curso, con orgullo: "¡Ahí está, profe, lo

pude mostrar en el navegador!", y enseguida varios estudiantes comenzaron a aplaudir de manera espontánea felicitándolo. El estudiante se sintió reconocido por sus compañeros, y ese gesto dio lugar a un momento especial, donde la alegría individual se volvió colectiva. Fue uno de esos momentos donde el aula vibra: cuando un pequeño logro técnico se transforma en un logro del grupo, y la clase cobra un sentido más pleno.

El tiempo se extendió más allá de lo planificado —la clase finalizó a las 14:10— debido a la necesidad de repetir explicaciones sobre cómo ejecutar correctamente el archivo con `node server.js` y desde qué ruta hacerlo. También surgió una pregunta interesante: si era posible cambiar el puerto predeterminado o dejarlo como variable de entorno, lo que dio lugar a una breve explicación sobre `process.env.PORT` y la práctica de definir puertos dinámicos para despliegue.

Todos los grupos lograron ejecutar su servidor local. Les comenté que debían guardar su trabajo para seguirlo en la próxima clase. Al despedirse, un estudiante me preguntó con entusiasmo si en la siguiente clase íbamos a trabajar con rutas donde le comenté que ese era nuestro siguiente tema. Ese cierre reafirmó el interés genuino que generó la actividad.

Desde mi rol docente, considero que la clase tuvo un desarrollo sólido en términos de planificación y ejecución. Los objetivos propuestos se cumplieron en gran medida, y la mayoría de los estudiantes lograron alcanzar la meta de levantar su primer servidor con Express. Sin embargo, también fue evidente que el ritmo de trabajo resultó exigente para algunos integrantes del grupo, especialmente para quienes requieren más tiempo para incorporar nuevos conceptos técnicos y llevarlos a la práctica.

Esta experiencia me permitió volver a reflexionar sobre la importancia de sostener una mirada amplia del grupo durante el dictado de la clase. A veces, en el afán de avanzar o de responder a la motivación de aquellos que captan los contenidos con mayor rapidez, uno puede caer fácilmente en acelerar el ritmo, sin contemplar que otros aún están transitando los primeros objetivos. Aprendí que es fundamental habilitar pausas estratégicas para observar, escuchar y ajustar la propuesta en función de las necesidades reales del grupo, promoviendo la ayuda entre pares y promoviendo la autonomía sin perder de vista los objetivos generales.

DIARIO DEL PROFESOR

Registro de clases

Institución: Escuela de gestión provincial

Curso: 5° año

Disciplina: Programación III

Fecha: 07/05/2025

Acerca de la institución

La institución está ubicada en un barrio ya consolidado desde hace años. Su presencia ha generado un movimiento inédito en la zona, donde la vida barrial no estaba acostumbrada a la circulación constante de jóvenes, docentes y actividades educativas cotidianas. Este cambio implicó, al principio, un proceso de adaptación mutua, pero con el tiempo se consolidó un vínculo de colaboración y pertenencia entre la escuela y su comunidad.

La escuela se caracteriza por ser abierta a la comunidad y por participar activamente en proyectos colectivos, encuentros interdisciplinarios y propuestas sociocomunitarias. Se desarrollan actividades integradoras como ferias, exposiciones y trabajos con organizaciones locales, lo que fortalece el lazo entre la formación académica y las realidades del entorno. En este sentido, la escuela se propone no solo formar estudiantes con competencias técnicas, sino también ciudadanos activos capaces de intervenir críticamente en su contexto.

En cuanto a la infraestructura, el edificio cuenta con cuatro aulas convencionales, destinadas al desarrollo de clases teóricas o grupales. A esto se suman tres laboratorios especializados, entre ellos un laboratorio de informática, que constituye el principal espacio para el dictado de clases relacionadas con programación y tecnologías digitales. Este laboratorio está equipado con computadoras individuales, conexión Wi-Fi de buena calidad, proyector y recursos que favorecen el trabajo colaborativo y autónomo de los estudiantes.

Todo esto da cuenta de una institución que apuesta por la innovación pedagógica, que valora el trabajo por proyectos, el uso intensivo de las TIC y que concibe el aprendizaje como un proceso colectivo, situado y significativo.

Acerca del tema

Eje: Backend con JavaScript

Contenidos y aprendizajes: Creación de servidores, manejo de rutas y controladores con Express.

Esta clase forma parte de una continuación directa de la secuencia didáctica iniciada en la clase anterior, dentro del eje de trabajo “Backend con JavaScript”. En la clase previa, los estudiantes habían logrado crear y ejecutar su primer servidor utilizando Express, comprendiendo conceptos fundamentales como servidor local, puerto, rutas básicas y el uso de comandos en la terminal.

En esta segunda instancia, el foco estuvo puesto en la organización del código del proyecto: específicamente, en la creación de una estructura de carpetas que permita separar responsabilidades dentro del backend. Para ello, se introdujo la carpeta routes, en la que los estudiantes comenzaron a implementar rutas específicas utilizando los métodos HTTP más comunes (GET, POST, PUT y DELETE).

Esta clase fue clave para comenzar a escalar la arquitectura del proyecto, dejando atrás el esquema de trabajo con un solo archivo (server.js) para avanzar hacia una estructura más modular y profesional. Además, se trabajó con base en el informe elaborado por los estudiantes en clases anteriores, donde ya habían definido los nombres de las rutas y controladores que formarían parte de su aplicación.

La actividad permitió reforzar la idea de escalabilidad y organización de proyectos reales, sentando las bases para el próximo paso: la implementación de los controladores. Esta continuidad progresiva en la secuencia favorece la apropiación paulatina de conceptos técnicos complejos, mediante una estrategia didáctica basada en objetivos intermedios y trabajo por proyectos.

Desarrollo de la clase

La clase comenzó puntualmente a las 12:00 hrs, en el espacio de informática nuevamente, cada uno en su estación de trabajo con sus propios compañeros de grupo para el proyecto, con un breve repaso sobre los conceptos claves de la clase anterior: ¿qué es un servidor?, ¿para qué sirve Express?, y ¿cómo se levanta un servidor local? Me llamó la atención cómo los estudiantes se acordaban de todo, incluso del comando exacto que debían escribir para ejecutar el servidor. Esta vez, en lugar de responder todos al mismo tiempo, varios levantaban la mano para participar, y las intervenciones se daban de forma más ordenada, lo que favoreció mucho la dinámica inicial. Se notaba que habían retenido bastante los conceptos y acciones de la clase anterior.

Luego de este repaso, les conté brevemente los objetivos que íbamos a lograr en esta clase: “Hoy vamos a empezar a organizar nuestras rutas en una nueva carpeta, para que

nuestro proyecto empiece a ser escalable”. Además de crear las diferentes rutas con los nombres de controladores que ustedes ya pensaron. A continuación, conecté mi notebook al televisor y comencé con la explicación. Esta vez propuse una dinámica diferente: invitarlos a acercarse con sus sillas y formar un semicírculo frente al televisor. La clase pasada, al dar la explicación desde el frente mientras ellos permanecían en sus computadoras, algunos se distraían o directamente no podían ver bien el código. Este pequeño cambio de disposición mejoró muchísimo la atención general.

Al volver a sus lugares, retomamos el archivo `server.js`. Les pedí explícitamente que no lo modificaran, sino que iban a crear una nueva carpeta llamada `routes`, donde comenzaríamos a implementar las rutas utilizando los métodos HTTP. Antes de seguir, les hice una pregunta al aire: “¿Cuáles eran los métodos HTTP que conocían?”. Un estudiante comentó con seguridad los cuatro más comunes: GET, POST, PUT y DELETE. Otro compañero, mencionó que existían otros como PATCH y OPTIONS, pero a modo de respuesta le comenté que probablemente no los usaríamos. Aproveché ese momento para felicitar ambos aportes, y aclarar que en este proyecto íbamos a trabajar con los más fundamentales, porque son los que usamos habitualmente para crear un CRUD.

Con el código proyectado, les mostré cómo estructurar una ruta básica y cómo nombrarla según el informe que habían creado clases anteriores. Cada grupo debía adaptar el nombre de sus rutas y preparar la estructura futura para los controladores, aún sin implementarlos. Como ya habíamos trabajado en pequeños objetivos, esta clase fue bastante más fluida. La mayoría de los estudiantes se organizó rápido y empezó a trabajar directamente. Había algunas dudas —cómo importar correctamente las rutas, cómo exportarlas, o cómo cambiar el nombre de la función—, pero fueron resolviéndolas entre compañeros, sin necesidad de tanta intervención docente.

A mitad de clase, recordé que en la siguiente clase íbamos a implementar los controladores. Les dije que hoy íbamos a dejar “preparado el escenario”, y que la clase siguiente íbamos a probar que cada ruta muestre un mensaje diferente según su función. Esta aclaración sirvió como motivación y cierre del bloque práctico.

En los últimos minutos de clase, retomé dos debates breves. Primero, un repaso sobre qué hicimos hoy y qué dificultades aparecieron. Segundo, una pregunta abierta al grupo: “¿Qué diferencias notan entre lo que hicimos hoy y lo que hicimos la clase pasada?”

Las respuestas fueron interesantes. Un grupo comentó: *“La clase pasada hicimos todo en un solo archivo, ahora ya estamos separando en carpetas”*. Otro agregó: *“Hoy no usamos tantas funciones todavía, pero ya estamos armando algo más organizado”*. Me gustó que saliera esa reflexión, porque era justamente uno de los sentidos de esta clase.

La clase finalizó a las 13:20. Al despedirse, varios estudiantes se mostraron tranquilos, y uno me preguntó si en la próxima íbamos a ver cómo hacer que esas rutas “respondan con datos”. Le respondí que sí, que empezábamos con los controladores.

Esta clase se vivió con mayor fluidez y autonomía por parte de los estudiantes. La base conceptual ya estaba más consolidada, lo que permitió avanzar sin tener que detenernos constantemente a repasar cada paso técnico. Algo tan simple como modificar la disposición del aula para que pudieran acercarse al televisor hizo una gran diferencia: se notó mayor

atención, menos distracciones y una comprensión más rápida del código que estábamos viendo. Eso reafirma para mí la importancia de cómo se comunica y desde dónde se enseñan los contenidos, no solo qué se enseña.

En cuanto a mi rol, pude tomar mayor distancia del paso a paso técnico para enfocarme en acompañar, observar y reforzar. Al no tener que intervenir tanto con correcciones, pude escuchar más las preguntas, los comentarios entre ellos y los debates que surgían naturalmente, como cuando distinguieron entre las rutas de hoy y el archivo único de la clase anterior.

Planilla de registro para el/la colega/coformador/a

1 Descripción general de la dinámica de la clase: organización y distribución de tiempos, actividades, recursos propuestos, etc.

- **Inicio de la clase:** Los estudiantes llegan a tiempo, se ubican en el aula de informática, el docente pasa lista, saluda y pide atención, solicitando que guarden sus celulares.
- **Repaso e Introducción:** El docente inicia con un repaso de los temas previos mediante un mapa conceptual proyectado en pantalla, vinculando la teoría con los ejercicios prácticos que los estudiantes realizan en las computadoras.
- **Desarrollo de la clase:**
 - Los estudiantes participan activamente, preguntan y resuelven problemas planteados por el docente.
 - El docente explica el tema, construye el mapa conceptual con pasos e instrucciones, da ejemplos, aclara dudas, y se asegura de que los estudiantes comprendan y puedan realizar las actividades.
 - El trabajo es en grupos, y el docente se desplaza por el aula, interactuando con los grupos y resolviendo dudas.
 - El docente utiliza la pantalla para proyectar información importante y pide a los estudiantes que se acerquen para prestar atención, además de dar consejos para resolver las actividades.
- **Cierre de la clase:** Se logra un buen ambiente de trabajo, con una actitud positiva y de apoyo por parte del docente. No se registran conflictos importantes.
- **Recursos:** Sala de informática, computadoras, pantalla de proyección, mapa conceptual.

2 Descripción en detalle de una o varias actividades (las que considere más relevantes).

- La actividad central es la creación de un servidor ("backend").
- Esta actividad se desarrolla de manera práctica en las computadoras, guiada por el mapa conceptual que el docente construye y explica.
- Los estudiantes resuelven ejercicios y situaciones problemáticas relacionados con la creación del servidor, con el apoyo del docente.

3 ¿Qué hace el/la profesor/a durante su desarrollo? ¿Qué tipos de intervenciones realiza? ¿Con qué finalidades las realiza?

• **Acciones:**

- Toma asistencia.
- Pide atención y el uso adecuado de los celulares.
- Realiza un repaso de los contenidos previos.
- Construye y explica un mapa conceptual.
- Plantea situaciones problemáticas.
- Da ejemplos.
- Responde dudas.
- Supervisa el trabajo en las computadoras.
- Se desplaza por el aula, trabajando con los grupos.
- Da "tips" y orientaciones.

• **Intervenciones:**

- Explicativas (del tema, del mapa conceptual, de las consignas).
- Interactivas (preguntas, planteo de problemas, diálogo con los estudiantes).
- De gestión del aula (pedir atención, mantener el orden).
- De apoyo individual y grupal (resolver dudas, dar orientaciones).

• **Finalidades:**

- Lograr la atención y el compromiso de los estudiantes.
- Facilitar la comprensión de los conceptos.
- Promover el aprendizaje activo y la participación.
- Guiar el proceso de trabajo práctico.
- Asegurar que todos los estudiantes alcancen los objetivos de aprendizaje.

4 ¿Qué hacen los/as estudiantes? ¿De qué maneras participan en la clase?

¿Cómo se vinculan entre sí y con el/la residente?

• **Acciones:**

- Llegan a clase y se ubican en el aula.
- Prestan atención a las explicaciones del docente.
- Participan activamente.
- Preguntan y plantean dudas.

- Resuelven las actividades en las computadoras.
- Trabajan en grupo.
- **Formas de participación:**
 - Responden a las preguntas del docente.
 - Plantean sus propias preguntas y dudas.
 - Proponen soluciones a los problemas planteados.
 - Colaboran con sus compañeros en el trabajo grupal.
- **Vinculación:**
 - **Con el docente:** Interactúan a través de preguntas, respuestas, y buscando su guía y apoyo.
 - **Entre sí:** Colaboran en los grupos de trabajo para resolver las actividades.

5

Aspectos significativos de la clase: acciones, frases textuales (de profesores/as y estudiantes). Pueden ser intercambios en torno al contenido.

- El docente pide a los estudiantes que dejen los celulares y presten atención.
- El docente utiliza un mapa conceptual como herramienta central de enseñanza.
- Se da una dinámica interactiva entre el docente y los estudiantes.
- El docente se asegura de que los estudiantes comprendan y puedan aplicar los conocimientos.
- El trabajo es en grupos.
- Se logra un buen clima de trabajo.

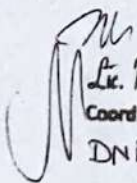
6

Descripción de conflictos, incidentes críticos, etc.

- No se detectan conflictos mayores o incidentes relevantes.
- El docente interviene si algún estudiante se distrae, pidiéndole que retome la atención.

7 Dudas, contradicciones, reflexiones que surgen durante, o después del desarrollo de la clase.

- **Reflexión sobre el aprendizaje activo:** La clase enfatiza fuertemente el aprendizaje activo mediante la participación estudiantil, el trabajo en grupo y las actividades prácticas. Esto es un aspecto muy positivo, ya que las investigaciones sugieren que el aprendizaje activo conduce a una comprensión más profunda y una mejor retención que la escucha pasiva.
- **El rol del mapa conceptual:** El mapa conceptual es una herramienta clave en esta clase. Parece proporcionar estructura y una ayuda visual para el aprendizaje.
- **El docente como facilitador:** El rol del docente se centra menos en la enseñanza y más en facilitar el aprendizaje. Se desplaza, responde preguntas y guía a los estudiantes.
- **Gestión y participación en el aula:** El profesor gestiona eficazmente la clase y mantiene un ambiente de aprendizaje positivo. Los estudiantes participan activamente y no hay interrupciones importantes.
- **Integración de la tecnología:** La clase se imparte en un laboratorio de informática, y la tecnología es fundamental para el proceso de aprendizaje, lo cual resulta apropiado para una clase de programación.


Lic. Ma. Florencia Rodríguez
Coordinadora ProA Las Varillas
DNI: 29.687.535

