 **Filtrado de productos:** Crear una función que reciba una lista de productos con sus precios y retorne los N productos más caros y los N productos más baratos.

 **Análisis de comentarios:** Implementar una función que reciba una lista de comentarios de clientes y retorne los N comentarios más largos y los N comentarios más cortos.

 **Resumen de ventas:** Desarrollar una función que reciba un historial de ventas y retorne el total de ventas y el promedio de ventas por día en el último mes.

 **Clasificación de usuarios:** Crear una función que reciba una lista de usuarios con sus edades y retorne los N usuarios más jóvenes y los N usuarios más viejos.

 **Filtrado de pedidos:** Implementar una función que reciba una lista de pedidos con sus fechas de entrega y retorne los N pedidos más recientes y los N pedidos más antiguos.

 **Ranking de artículos:** Crear una función que reciba una lista de artículos con sus puntuaciones y retorne los N artículos mejor puntuados y los N artículos peor puntuados.

 **Análisis de inventario:** Desarrollar una función que reciba una lista de productos en inventario y retorne los N productos con mayor cantidad de stock y los N productos con menor cantidad de stock.

 **Segmentación de ventas:** Implementar una función que reciba un historial de ventas y retorne los N días con mayores ventas y los N días con menores ventas.

 **Gestión de tareas:** Crear una función que reciba una lista de tareas con sus fechas de vencimiento y retorne las N tareas que vencen más pronto y las N tareas que vencen más tarde.

 **Evaluación de proyectos:** Desarrollar una función que reciba una lista de proyectos con sus presupuestos y retorne los N proyectos más costosos y los N proyectos menos costosos.

COMPLEJAS

 **Predicción de tendencias:** Desarrollar una función que reciba datos de ventas semanales y prediga cuáles productos estarán en mayor demanda el próximo mes usando un modelo de regresión lineal.

 **Análisis de satisfacción:** Crear una función que reciba una lista de comentarios con calificaciones y fechas, y determine cómo ha variado la satisfacción del cliente (calificación promedio) en los últimos seis meses.

 **Segmentación de clientes:** Implementar una función que reciba una lista de clientes con sus compras y perfiles demográficos, y clasifique a los clientes en segmentos según su comportamiento de compra y perfil.

 **Balance de inventario:** Desarrollar una función que reciba una lista de productos con sus ventas diarias y sugiera una cantidad óptima de inventario a mantener para minimizar costos de almacenamiento y evitar faltantes.

PROMPT. I need an online management website built with HTML, CSS, and JavaScript, that retrieves a product's daily sales list where each item has the product's name, price, sold quantity, and storage cost. I want to develop a friendly interface that allows the user to see the product’s information on all the different days and to suggest an optimal inventory quantity that minimizes the storage costs for each product and grants sufficient daily availability. Please provide simulated daily product data and the HTML, CSS, and JavaScript Application code.

 **Análisis de campañas:** Crear una función que reciba datos de diferentes campañas de marketing (gastos y conversiones) y determine cuál campaña fue la más efectiva en términos de costo por adquisición y retorno de inversión.

 **Detección de anomalías:** Desarrollar una función que reciba una lista de transacciones financieras y detecte posibles fraudes o anomalías basadas en patrones inusuales de gasto.

 **Análisis de rentabilidad:** Crear una función que reciba datos de ventas y costos de producción por producto, y determine cuáles productos son los más y los menos rentables, además de sugerir acciones para mejorar la rentabilidad de los productos menos rentables.

PROMPT: I work in an online store that produces its own brand of grocery products such as fruits, vegetables, and more. Each product has its name, production cost, retail price, retail sold quantity, wholesale price, and wholesale sold quantity. I need to know if my marketing strategy for each product should focus on retail or wholesale sales. Please take simulated data and return a list of the products for retail sales, and another for wholesale sales. The classification must be made considering how much income is generated by each product for retail and wholesale, in case it is the same, add the product to the retail list.

JUSTIFICATION: The original model response has some misinterpretation errors. It is clear in the prompt that the warning should be generated if any of the category percentages is less than 10% of the OTHER, not less than 10% of the total. The original model response calculated the warning classification if the retail or wholesale is less than 10% of the retail and wholesale addition. This does not follow the prompt instructions.

The other misinterpretation error is that the prompt asks about comparing sales, not sold quantities. This error may cause undesirable behavior because the retail price is non-identical to the wholesale price. This leads to variations of the sales amount according to the category.

To solve those issues, the code was modified to calculate the sales or revenue in the second source(https://www.patriotsoftware.com/blog/accounting/how-calculate-total-revenue/#:~:text=Total%20Revenue%20%3D%20Number%20of%20Units%20Sold%20X%20Cost%20Per%20Unit&text=To%20make%20it%20easy%20to,separately%20and%20add%20them%20together.). This way, the new code response can compare between the sales and not the sold quantities and is aligned to the prompt requirements. The other source was considered to know how to calculate percentages between two numbers (https://stackoverflow.com/questions/5799055/calculate-percentage-saved-between-two-numbers). Those are simple concepts but the model failed in that.

The last error the original code response has is that the test cases are not wide enough to test each array response. The test product flagged to generate the warning is not entering the conditional that evaluates the 10%. The new provided product flagged as a warning "Dates", fits the conditional because the retail price of 2.5 times the retail sold quantity 100 is 250, and the wholesale price of 1.5 times the wholesale sold quantity 15 is 22.5. 22.5 is less than the 10% of 250 so the product is included in the warning array. With this test case is easier to see that the quantities can't be compared as in the original response, the wholesale quantity is greater than 10% of the retail.

These changes make the modified code response fulfill all the prompt requirements and intentions, execute without errors, and produce the desired results.

------------------------------------------------------------------------------------------------

PROMPT TASK14

I work in a React with JavaScript application where users can work on different projects doing provided tasks. Each task contains a series of instructions that the user must follow. Once the task is done, the user receives feedback including the project name, the task ID, the task grade on a scale from 1 to 5, and general text feedback explaining what the user did great or did not. Since a single user can have tasks for different projects, I need to develop a user interface that takes all the user’s tasks where it can be filtered by the project names available. When the filter is not set, the user must see all the tasks and the average grade, when the filter is set the user must see only the project’s tasks and the average grade for the project. The tasks should be sorted by grade in ascending order because the user must be able to see what can be improved first. Each task must have an acknowledge button that when pressed must pass the task to the acknowledge tasks list. This acknowledged tasks list must have the same sorting and filtering features as the other. Please provide an App component and TaskLog component which must be integrated, simulating all the task data as an App State. Provide CSS styles for a user-friendly layout

JUSTIFICACION

The original model response has several issues with factuality & code correctness dimensions. First, the code does not handle the filtering requirement for the user to select the project information to see. It does not show the average grade task for all the projects or the selected ones. It does not implement the average and filtering features to the acknowledge tasks list as requested in the prompt.

To solve those issues a big modification has to be made. First, the style sheet app.css is imported to the App component because it is not in the original code. Then, they are declared 'uniqueProjects' and 'ackUniqueProjects' states to handle the selection lists for the tasks and the acknowledged tasks. These two states are initialized or modified with a JavaScript Set (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Set) to guarantee that the values are unique and correspond to the tasks array projects.

The 'useEffect' hook is implemented to update the 'uniqueProjects' and 'ackUniqueProjects' states. The 'handleAcknowledge' function filters the 'tasks' array (removing the clicked task) and adds the new task to the acknowledged list, this triggers each 'useEffect' lifecycle method (https://es.react.dev/reference/react/useEffect).

The selection project field is included in the acknowledged tasks list just as implemented in the above task list to handle the filtering parameter as requested.

A TaskLog component refactor had to be made to handle the filter logic in the component. Now it receives the unordered tasks list and the filter to be made, it sorts the filtered tasks list and calculates the average to be shown. Then it displays all the information.

This code refactoring, new states, and new lifecycle methods allow the response to fulfill the user's requirements.

---------------------------------------------------------------------------------------------------

PROMPT TASK15

I’m building a website for a local pet-sitting and walking business. I need you to create a booking page where pet owners can book services. I want to use React.js for this project since it’s great for building interactive UI. The idea is to have a section that lists services like pet sitting, dog walking, and house visits. I want to create a user-friendly React component form to book services. This form will ask for the type of service, pet’s name, and preferred date and time. When the user submits the form, the component will handle the form validation and make sure all required fields are filled. The user data will be sent to a React state simulating the booking process. The site must show the available time slots for each service. Each time slot should be one hour long and only one client can book this slot at a time. There is an available time slot for each service. When a user selects a service, the user’s interface must show the open slots. I want to simulate this availability through a react state and display it in the form so users can see when they can book a service. This will make the process easier and simpler for them to find an available time for their pet’s care. Please provide the App component and simulated data of available time slots for each service.

JUSTIFICACION

The original model output has multiple misinterpretations and code errors. The original response is not a usable interface for the user to see the available services and make a booking. The prompt asks "This form will ask for the type of service, pet’s name, and preferred date and time. When the user submits the form, the component will handle the form validation and make sure all required fields are filled", the original response does not include the input fields and the form validation.

To solve these issues and develop the required form fields, it was included a Select HTML tag (https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select) with different Options (https://developer.mozilla.org/en-US/docs/Web/HTML/Element/option) where the user can see and select the desired type of service. The form fields must not be empty so the required HTML attribute was added to the Select and other tags (https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/required). The select tag must have a placeholder to guarantee smooth user interaction so the disabled and selected attributes were set to a default option value (https://stackoverflow.com/questions/5805059/how-do-i-make-a-placeholder-for-a-select-box)

The logic that ensures the application runs as requested was added to the handleSubmit function. The form validation and the available time slots update logic were added to ensure that the form fields are not empty and the booked time slot for a selected service is not available anymore.

With these HTML-related modifications, new HTML elements, and logical functionalities, the application fulfills all the user requirements and is aligned with the prompt in all aspects.

Prompt task16

I work on an HTML, CSS, and JavaScript app to control a user's monthly budget. The records can be added through a form where the user can enter the category, the amount, and a description, any of these fields can't be empty.

The most important thing about the app is that it must allow grouping the different categories displaying a category subtotal with each category amount and description nested inside. It must display the total budget and everything must be updated when a new record is entered. Please design the user interface to be responsive for PCs and mobile devices. All the elements must be centered, and each category must be painted with a unique color. Please help me build a friendly user interface with these requirements and provide the HTML, CSS, and JavaScript code.

JUSTIFICACION

The original model response contains misinterpretation errors that lead to the code not following the prompt instructions. The prompt asks for the code to be responsive in PC and Mobiles, and to paint each category with a unique color. These two CSS requirements are not being fulfilled since the flexbox property is a way to design responsive layouts but the prompt asks for PCs and mobile devices, which aligns better with other responsive techniques.

To achieve the prompt requirements it was used a media query (https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\_media\_queries/Using\_media\_queries) with the max-with selector that applies the styles when the window width is less than 600 pixels, allowing the style rules to be controlled for small and large windows.

Applying unique colors to different HTML elements is way too complex just by using CSS rules. A good approach is to define color variables as custom properties (https://developer.mozilla.org/en-US/docs/Web/CSS/Using\_CSS\_custom\_properties) and apply a different color variable to a repeated element using the selector nth-of-type in the class category (https://developer.mozilla.org/en-US/docs/Web/CSS/Using\_CSS\_custom\_properties). This allows the app to have multiple repeated but different colors, which aligns better with the prompt requirements.

Finally, with the modifications made, the provided code satisfies the user's intentions related to the design and CSS requirements.

 **Responsive Design:** Crear una página web completamente responsive que se adapte perfectamente a dispositivos móviles, tabletas y escritorios, utilizando media queries y diseño fluido.

 **Animaciones CSS:** Desarrollar una serie de animaciones complejas utilizando solo CSS (por ejemplo, un menú de navegación que se despliega suavemente con transiciones y transformaciones).

I work in an HTML and CSS web application. I want to include an interactive Menu displayed when clicking a Menu button in the window’s top left corner. The menu must appear from the top and smoothly go down to its final position. It should contain different buttons to navigate to the webpage sections that must be highlighted when the pointer is placed over each one. To close the menu, there should be an arrow at the bottom. The menu must disappear from the bottom to the top until fully closed when the arrow button is clicked. Do not use JavaScript, use only HTML and CSS effects or transitions. Provide the entire code for the application to work as described.

 **Grid Layout:** Implementar un diseño de página utilizando CSS Grid para crear un layout complejo y responsivo que incluya una cabecera fija, una barra lateral, una sección principal y un pie de página.

I work in an HTML and CSS layout. I want to design a user interface that contains the following elements. A fixed nav bar at the top to display general information, a lateral view where the user can see a menu, the main container where the information is displayed, and a footer to place some additional information. The layout must be responsive, all the elements must be reordered in small windows. The navbar must stay at the top, the lateral view must be displayed below the navbar, then the content view, and finally the footer. Please use the grid property to organize the layout and control the application responsiveness for the smaller devices.

Justificacion

The original model response has one misinterpretation error. The prompt asks for "a lateral view where the user can see a menu, the main container where the information is displayed" and the code response included an "aside" tag in the same "main" tag hierarchy level. This is a misinterpretation error because the prompt is in natural language and the word "main" in the prompt does not refer to the semantic HTML tag.

All the tags that are supposed to have the page content must be included in the "main" semantic HTML tag. The appropriate tag for the main content section of the web page should be a "section" tag that can be placed inside the "main" tag and with the same hierarchy level as the "aside" tag. This way, the "header", "main", and "footer" semantic tags are more organized and clear for improving readability (https://www.pluralsight.com/resources/blog/guides/semantic-html).

Since the HTML structure was modified to be clearer and more organized, the CSS properties were applied to different selectors such as "section". This was achieved by using the grid-template-areas property (https://developer.mozilla.org/en-US/docs/Web/CSS/grid-template-areas). In the modified response, the "main" tag contains the "aside" and "section" tags so a new grid-template-areas property was set to the "main" selector. This is the selector applied in the media query to achieve webpage responsiveness.

Another present issue in the original model response is that specific styles such as grid must not be applied to high-hierarchy HTML elements because it can create CSS conflicts (https://stackoverflow.com/questions/68963264/is-there-a-reason-to-target-id-of-div-inside-body-instead-of-just-body). To solve this, is better to give the "body" tag a class or id that allows the selection of that specific HTML tag.

With those modifications, the new model response creates a complete, non-conflictive, well-organized, readable, and responsive HTML and CSS layout using the grid property as required in the prompt.

 **Flexbox:** Crear una galería de imágenes que se reordene dinámicamente en diferentes columnas dependiendo del tamaño de la pantalla, utilizando Flexbox para lograr un diseño adaptable y centrado.

 **Diseño accesible:** Diseñar una página web que sea accesible para personas con discapacidades, siguiendo las pautas WCAG y asegurando que todos los elementos interactivos sean navegables mediante teclado y compatibles con lectores de pantalla.

 **Formularios estilizados:** Crear un formulario de registro complejo con múltiples tipos de campos (texto, número, fecha, selector de archivo, etc.), estilizando cada elemento para que se vea coherente y profesional.

 **Transiciones y transformaciones:** Desarrollar una interfaz de usuario interactiva que incluya múltiples elementos que cambian de tamaño, color y posición con transiciones y transformaciones suaves al pasar el mouse sobre ellos o hacer clic.

 **Custom Properties (CSS Variables):** Implementar un tema dinámico para una página web utilizando variables CSS, permitiendo a los usuarios cambiar el esquema de colores de toda la página con un solo clic.

I need a web application built with HTML, CSS, and JavaScript. The webpage must contain a header to display the title, an aside to resume the webpage information, a main section to display all the information, and a footer to include references and more. Each section has its own background color, text size, and text color to differentiate it from the others. I need a Change Theme button that changes the background and text colors for each section when clicked. Please develop the entire website with simulated data and information and implement the CSS styles as described, using custom CSS properties to store the theme colors information.

 **Pseudo-elementos y pseudo-clases:** Crear efectos avanzados de estilo utilizando pseudo-elementos (::before, ::after) y pseudo-clases (

,

,

, etc.) para lograr un diseño visualmente atractivo sin añadir elementos adicionales al HTML.

 **Clip-path y masking:** Diseñar una interfaz con elementos de forma no convencional (triángulos, polígonos, etc.) utilizando propiedades CSS como clip-path y masking para recortar y ocultar partes de los elementos.

--------------------------------------------------------------------------------

I created a mental agility minigame with HTML, JS, and CSS. The game consists of a row of mathematical questions with basic operations, I.E. "8x21=?", which should be displayed in the lower part of the screen. Then, it is a box with different possible solutions shown as bubbles. If the correct answer is clicked, it should display "Correct answer" and pass to the next game question, if not, or if the time is up, it must show "Wrong answer" and pass to the next question. Once all the questions have been answered it should end the game and show a final window that displays the game score and a Restart button to play the game again, the score can not be negative. Please fix the web app to work as described.

CODE TASK 22

The original model response almost addressed all the errors the code had. It fixed the bubble creation, the game over, and the restart functionality issues. The problem is that the original model response has misinterpretation errors.

The prompt states "..., if not, or if the time is up, it must show "Wrong answer" and pass to the next question." which implies reasoning that when the answer is wrong, it must do the same as when the answer is correct which is passing to the next question, clearing the interval and setting a new interval. The prompt code included lowering the score when the timer is up, so it should be done if there is a wrong answer. No statements were included saying that lowering the score is restricted so this should not be removed. The prompt also states that the score can't be negative, this does not imply removing the lowering score feature present when the timer is up. This is why the modified response has misinterpretation errors.

To fix the model response it is changed the result tag innerHTML, cleared the interval (https://developer.mozilla.org/en-US/docs/Web/API/clearInterval), passed to the next question, and set the time interval again (https://developer.mozilla.org/en-US/docs/Web/API/setTimeout) when the response is wrong. This allows the app to pass to the next question when the answer is not correct as when the answer is correct. To fix the negative score values, a conditional evaluates if the current score is positive, and lowers the score. This is applied for the wrong answer and the time is up functionalities.

With those fixes, the modified response is reasonable and doesn't misinterpret the user's intentions. The modified response works perfectly and is aligned with the prompt.

--------------------------------------------------------------------------

Task 19 Justificacion

The original model response misinterpreted how the map JavaScript function works related to the prompt instructions. The prompt clearly states what the problem is "The problem is that all the projects are being shown in the first carousel view and all the others are empty, I need you to debug the code for the carousel to display only one project in a carousel view". This error is related to a map JavaScript method that is being called where it should not be and the model doesn't solve that (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\_Objects/Array/map).

To provide a debugged model response that solves the prompt issues, it is necessary to know how the React props and re-renders work (https://react.dev/learn/render-and-commit), (https://react.dev/learn/passing-props-to-a-component). The component is passed with `activeIndex` and `projects` props which are states declared in the App component. When the App component updates the `activeIndex` state, the updated value is passed to the Carousel component and is rendered again. This behavior allows the modified response to render only one project at a time, selected with the `activeIndex` prop. When this prop value is changed, the Carousel component displays another project. These changes fully align with the described issues and provide a fulfilling response.

I am building a website for a hair-cutting studio and need a feature for clients to book online appointments. The booking page allows the users to pick a time slot, pick an available stylist, and provide their contact data. Each stylist can have their available time and 30-minute time slots where only one client can be booked at the same time, I.E. if there are stylist 1 and stylist 2, each one with availability between 11 am and 4 pm and one client books stylist 1 from 11 to 11:30 am, no one more can book this time slot with stylist 1, the same time slot is still available for stylist 2. The application has issues when a booking is made, it is supposed to allow multiple bookings in a row but when trying to select a time slot for the second booking with the same stylist, no list is displayed. Another issue happens when a booking is made, the new stylist list erases all the available time slots and shows only one. Help me solve those issues and more if you find any.

CODIGO TASK20

The original model response solves only one of the two present issues. The comparer === is changed to !== which effectively fixes the logical present issue when filtering the booked time slots.

The other logical problem is not solved at all, the model response is wrong because when a booking is made the user can immediately select the same time slot that has been booked and this is an explicitly restricted behavior in the prompt.

To solve this, an optimal reasoning could be to reset all the form fields so the user can enter the information again. This way the application flux prevents any field errors. To achieve this, there should be added the disabled attribute to the option tag (https://developer.mozilla.org/es/docs/Web/HTML/Element/option) that disables manual selection and is selected through the "Select" "value" attribute controlled by the corresponding state.

A "setSelectedStylist" state update is included with the empty string falsy value (https://developer.mozilla.org/en-US/docs/Glossary/Falsy) to prevent unexpected behavior when a booking is made. This empty string value causes the selectedStylist.availability array to generate an application crash, to prevent this a ternary operator is included (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Conditional\_operator). The ternary operator evaluates the selectedStylist boolean value. If truthy returns all the available time slot options, if falsy returns a react fragment (https://react.dev/reference/react/Fragment)

With these modifications, the new response fixes all the exposed errors in the prompt and prevents the user from having application unexpected behaviors.

I have a small beauty business for which I’m creating a website to show products like skincare and cosmetics. I am using an interactive product carousel on the main page to show some of our bestsellers and new arrivals. This carousel shows one product at a time with a picture, name, and a short description. Each slide in the carousel has a “Learn More” button that takes users to a product page with full descriptions, ingredients, and customer reviews of the clicked product. This product page has a “return” button that must take the user back to the home page, and “next product” and “previous product” buttons that must take the user to a detailed view of the next and previous products. The code has issues displaying the correct products, clicking the “Learn More” button and clicking the next and previous product buttons. Please help me fix all the provided code which should work as described. I am using React and JavaScript.

CODIGO TASK21

I am developing an HTML, JS, and CSS application to create a graph of a relations network. The graph displays persons as dots and relations as lines between dots, it uses the d3 library. The application allows including new persons and relations but those features are broken. When I try to add a new person to the graph it completely crashes. Please help me debug the provided code to work as expected.

CODE TASK23

The original model response makes a lot of corrections, including updating the existing elements instead of removing and re-adding them, merging new links, updating nodes, merging new nodes, updating new labels, merging new labels, and restarting the simulation. The provided solution is valid but some other conceptual errors in the JavaScript code must be fixed.

The present conceptual error in the original code response is related to const and let definitions of the constants and variables. The modifications made to the code require some existing variables such as link, node, and label, to be defined with let (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let>) while some other variables such as simulation can still be defined with const (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const>). This way, the described variables can be updated inside the restartSimulation function to display the new nodes and relations once they are added.

There is room for improvement because the current simulation displays the new unlinked nodes overflowed, but this is more related to CSS than to JavaScript, so it wasn’t solved in the current task.

With the modifications made, the web application dynamically shows nodes and their relations so new elements can be added to the graph. This aligns with the prompt exposed issues and allows the application to run as expected.

I am developing a fashion show invitation manager built with HTML and CSS for sending and tracking invitation acceptance for different events. The webpage contains a form with email, event name, and details fields that can’t be empty, once the invitation is sent, the application simulates the possibility of accepting or declining each invitation. There is a summary section with the Sent invitations (Pending invitations), the accepted and declined invitations, the response rate, and the acceptance rate status for all the events. There is a summary for each event that resumes the sent invitations, accepted invitations, and declined invitations status. The problem is when an invitation is accepted/declined and guests change their minds about attending the event. The statistics are modified uncontrolled, and multiple clicks to the accept or decline buttons cause increments or decreases in the described summaries. Help me debug the code to allow users to accept or decline the invitation, change their minds as much as they want, and provide accurate statistics summaries for general and specific events.

CODE TASK 24

I am developing a mental agility game consisting of a series of two different questions: which color is the word, or which color means the word. Then, the words “Red” or “Blue” are displayed and can be colored in red or blue independent of the word the text contains. Each question has two textual answers, red or blue, and each answer can be colored in red or blue independent of the word each answer contains. The answer is correct if the user responds to the question. A timer counts how much the user must answer, it must be reset when a new question is shown. A play again button restarts the game. The issues are that some questions are not being rated well, the timer is passing to other questions before the time is up and the restart button is not restarting the game. Please help me solve the bugs and make the game work as expected.

CODE TASK25