React

TechGuide - Alura, FIAP e PM3

Nivel 1

HTML - Fundamentos:

- HTML es un lenguaje de marcado que define la estructura de su contenido. HTML consta de una serie de elementos que se utilizan para que se vea o actúe de cierta manera. Las etiquetas de archivos adjuntos pueden vincular una palabra o imagen a otro lugar, pueden poner palabras en cursiva, pueden hacer que la fuente sea más grande o más pequeña, etc.
- Aprender qué etiquetas son necesarias para HTML básico
- Crear de un párrafo de texto
- Mostrar una imagen
- Conocer la diferencia entre 'h1', 'h2', 'h3', etc.
- Crear de un texto con hipervínculo
- Crear un formulario con campos relevantes
- Crear de una lista ordenada o desordenada de elementos
- Crear de una lista de elementos en una lista desplegable
- Vincular a un archivo CSS
- Crear una tabla
- Adicionar ID y clases

Contenidos

- `Web` HTML: Lenguaje de etiquetas de hipertexto (https://developer.mozilla.org/es/docs/Web/HTML)
- `Web` W3Schools: Formato de texto HTML (https://www.w3schools.com/html/html_formatting.asp)
- `Web` W3Schools: Bloque HTML y elementos en línea (https://www.w3schools.com/html/html_blocks.asp)
- `Web` W3Schools: Elementos y técnicas de diseño HTML (https://www.w3schools.com/html/html_layout.asp)
- `Web` W3Schools: Listas HTML (https://www.w3schools.com/html/html_lists.asp)
- `Web` W3Schools: Tablas HTML (https://www.w3schools.com/html/html_tables.asp)
- `Web` W3Schools: Formularios HTML (https://www.w3schools.com/html/html_forms.asp)
- `Web` MDN Web Docs: Conceptos básicos de HTML (https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)
- `YouTube` ¿Qué es HTML? - 10 cosas que debes saber (https://www.youtube.com/watch?v=tPzq8lufGxE)
- `YouTube` Aprende HTML en 15 Minutos (https://www.youtube.com/watch?v=mNbnV3aN3KA)
- `YouTube` HTML - Ejemplos Prácticos básicos (https://www.youtube.com/watch?v=NMh94GBeCCQ)

Contenidos Alura:

- `Artículo` Por una web más rápida: 26 técnicas de optimización de Sitios Web (https://www.aluracursos.com/blog/por-una-web-mas-rapida-26-tecnias-de-optimizacion-de-paginas-web)
- `Artículo` La semántica en HTML5 (https://www.aluracursos.com/blog/la-semantica-en-html5)
- `Artículo` HTML, CSS y Javascript, ¿cuáles son las diferencias? (https://www.aluracursos.com/blog/html-css-javascript-cuales-son-las-diferencias)
- `Artículo` Formulario con form validation de HTML5 (https://www.aluracursos.com/blog/formulario-con-form-validation-de-html5)
- `Artículo` Empezando con el desarrollo web Front-end (https://www.aluracursos.com/blog/empezando-con-el-desarrollo-front-end)
- `Artículo` Desde cero hasta programador front-end (https://www.aluracursos.com/blog/desde-cero-hasta-programador-front-end)
- `Artículo` VisualStudio Code: instalación, teclas de acceso directo, plugins e integraciones (https://www.aluracursos.com/blog/visualstudio-code-instalacion-teclas-de-acceso-directo-plugins-e-integraciones)
- `Artículo` Llenando un formulario HTML automáticamente con AJAX (https://www.aluracursos.com/blog/llenando-un-formulario-html-automaticamente-con-ajax)
- `Artículo` Vinculando elementos con HTML5 (https://www.aluracursos.com/blog/vinculando-elementos-con-html5)
- `Artículo` ¿Qué es HTML y sus tags? Estructura básica (https://www.aluracursos.com/blog/html-y-sus-etiquetas)
- `Artículo` ¿Qué es HTML y sus tags? Elementos inline (https://www.aluracursos.com/blog/que-es-hmtl-y-sus-etiquetas-parte2-elementos-inline)
- `Artículo` ¿Qué es HTML y sus tags? Elementos a nivel de bloque (https://www.aluracursos.com/blog/que-es-html-y-sus-tags-parte-3)
- `Artículo` ¿Qué es HTML y sus tags? Elementos de un formulario (https://www.aluracursos.com/blog/que-es-html-parte-4)
- `Artículo` ¿Qué es HTML y sus tags? Atributos de los elementos. (https://www.aluracursos.com/blog/ques-es-html-parte-5-atributos)
- `Curso` HTML & CSS: Crea páginas increíbles con tecnologías web (https://app.aluracursos.com/formacion-html-css)

CSS - Fundamentos:

- Las hojas de estilo en cascada (CSS) son un lenguaje de hoja de estilo usado para descubrir una presentación de un documento escrito en un lenguaje de marca, como HTML o XML. O CSS puede ser usado para estilizar textos de documentos muy básicos — por ejemplo, para alterar a cor e o tamanho de cabeçalhos e links. Ele pode ser usado para criar um layout — por ejemplo, transformando uma única columna de texto en um layout com uma área de conteúdo principal e uma barra lateral para información relacionada. Pode até ser usado para efectos como animação.
- Aprender a estructura visual de uma página, con 'margem' e 'preenchimento'
- Estabelecer o tamanho com 'larga' e 'altura'
- Aprender sobre la posición de un elemento ('estático', 'relativo' o 'absoluto')
- Aprender sobre la exposición de un elemento ('block', 'inline', 'inline-block')
- Aprender a posicionar imágenes en relación con el texto

- Aprender sobre alinhamento
- Aprender sobre estilo de fuente
- Aprender las diferencias y ventajas de usar las diferentes unidades de medida en CSS (%, relativo, etc.)
- Conectar-se a los elementos (IDs, clases) de un archivo HTML
- Cambiar las características de un elemento cuando se pasa el ratón sobre él
- Aprender a dimensionar cajas
- Aprender Flexbox
- Grado de aprendizaje

Contenidos

- `Web` CSS (https://developer.mozilla.org/es/docs/Web/CSS)
- `Web` W3Schools: Introducción a CSS (https://www.w3schools.com/css/css_intro.asp)
- `Web` W3Schools: Sintaxis CSS (https://www.w3schools.com/css/css_syntax.asp)
- `Web` W3Schools: CSS Selectores (https://www.w3schools.com/css/css_selectors.asp)
- `Web` W3Schools: CSS Box Model (https://www.w3schools.com/css/css_boxmodel.asp)
- `Web` W3Schools: Unidades CSS (https://www.w3schools.com/css/css_units.asp)
- `Web` MDN Web Docs: Comenzando con CSS (https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started)
- `Web` MDN Web Docs: Cascada, especificidad y herencia. (https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance)
- `Web` MDN Web Docs: Propiedades de visualización (https://developer.mozilla.org/en-US/docs/Web/CSS/display)
- `Web` MDN Web Docs: Conceptos básicos de flexbox (https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)
- `Web` MDN Web Docs: Conceptos básicos de Grid Layout (https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)
- `YouTube` Que rayos son las variables en CSS y como se utilizan. (https://www.youtube.com/watch?v=nJO1jjKUrol&ab_channel=FalconMasters)

Contenidos Alura:

- `Artículo` Bootstrap ¿Qué es, cómo y cuándo utilizar? (https://www.aluracursos.com/blog/bootstrap-que-es-como-y-cuando-utilizar)
- `Artículo` Comprenda la propiedad Position CSS (https://www.aluracursos.com/blog/comprenda-la-propiedad-position-css)
- `Artículo` CSS: Grids y tablas con responsividad en la Web (https://www.aluracursos.com/blog/CSS-Grids-y-tablas-con-responsividad-en-la-Web)
- `Artículo` Puntos de ruptura en el diseño responsivo (https://www.aluracursos.com/blog/puntos-de-ruptura-en-el-diseno-responsivo)
- `Artículo` Flexbox CSS: Guia Completo, Elementos y Ejemplos (https://www.aluracursos.com/blog/flexbox-css-guia-completo-elementos-y-ejemplos)
- `Artículo` ¿Por qué usar 'em' en su CSS hoy? (https://www.aluracursos.com/blog/por-que-usar-em-en-tu-css-hoy)
- `Artículo` Tu código CSS puede ser más limpio, flexible y reutilizable. (https://www.aluracursos.com/blog/tu-codigo-css-puede-ser-mas-limpio-flexible-y-reutilizable)

- `Artículo` Creando Layouts con Css Grid Layout (https://www.aluracursos.com/blog/creando-layouts-con-css-grid-layout)
- `Artículo` Creación de componentes CSS con el estándar BEM (https://www.aluracursos.com/blog/creacion-de-componentes-css-con-el-estandar-bem)
- `Artículo` Empezando a organizar tu CSS (https://www.aluracursos.com/blog/empezando-a-organizar-tu-css)
- `Artículo` Reset CSS: Qué es, Ejemplos, Cómo Crear y Utilizar (https://www.aluracursos.com/blog/reset-css-que-es-ejemplos-como-crear-y-utilizar)
- `Artículo` ¿Cómo lidiar con los límites de resolución en sitios responsivos? (https://www.aluracursos.com/blog/como-lidiar-con-los-limites-de-resolucion-en-sitios-responsivos)
- `Artículo` Cómo organizar el CSS en tu proyecto (https://www.aluracursos.com/blog/como-organizar-el-css-en-tu-proyecto)
- `Artículo` CSS: animaciones en Transition y Animation (https://www.aluracursos.com/blog/css-animaciones-de-transition-y-animation)
- `Artículo` Guía de Unidades en el CSS (https://www.aluracursos.com/blog/guia-de-unidades-en-css)
- `Artículo` Cambiando CSS con JavaScript (https://www.aluracursos.com/blog/cambiando-css-con-javascript)
- `YouTube` Animando un texto en HTML y CSS (https://www.youtube.com/watch?v=j71g5TiMA-M&ab_channel=AluraLatam)
- `YouTube` Box Model y Box sizing #AluraMás (https://youtu.be/ts9qfCKamKg)
- `YouTube` Consejos de CSS FlexBox para comenzar #aluramás (https://youtu.be/EB4vWLzfVcI)
- `YouTube` Como utilizar el Display block, inline, inline-block #aluramás (https://youtu.be/AG2QssLpQzI)
- `YouTube` ¡Domina CSS Flexbox con FlexboxFroggy! Aprende jugando en este desafío divertido (https://youtu.be/MXPhyN5t0uQ)
- `Curso` Flexbox: Posicione elementos en la pantalla (https://www.aluracursos.com/curso-online-flexbox-posicione-elementos-pantalla)
- `Curso` CSS Grid: Simplificando layouts (https://www.aluracursos.com/curso-online-css-grid-simplificando-layouts)
- `Curso` Arquitectura CSS: descomplicando los problemas (https://www.aluracursos.com/curso-online-arquitectura-css-descomplicando-los-problemas)

Javascript - Fundamentos:

- Javascript es el lenguaje de programación más popular del mundo y es una de las tecnologías centrales de la World Wide Web, junto con HTML y CSS. Tiene escritura dinámica, orientación a objetos basada en prototipos y funciones de primera clase. Es un paradigma múltiple que admite estilos de programación imperativos, funcionales e impulsados por eventos.
- Conocer los tipos primitivos
- Declarar variables, considerando la diferencia entre 'var', 'let' y 'const
- Uso de estructuras condicionales ('if', 'else')
- Conocer los operadores de asignación y comparación ('=', '==', '===')
- Uso de estructuras de repetición y bucles ('while', 'for')
- Usar funciones, pasar parámetros y argumentos
- Manipulación de arreglos y listas

- Aprender el concepto de Orientación a Objetos
- Realizar un CRUD
- Obtener datos de una API
- Hacer llamadas asincrónicas usando 'Async/Await', 'Promise', etc.

Contenidos

- `Web` MDN Web Docs: JavaScript (https://developer.mozilla.org/es/docs/Web/JavaScript)
- `Web` MDN Web Docs: Guía de JS (https://developer.mozilla.org/es/docs/Web/JavaScript/Guide)
- `Web` MDN Web Docs: Gramática y tipos (https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Grammar_and_types)
- `Web` W3Schools: Comparación de JavaScript y operadores lógicos (https://www.w3schools.com/js/js_comparisons.asp)
- `Web` MDN Web Docs: Expresiones y operadores (https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Expressions_and_operators)
- `Web` MDN Web Docs: Array (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array)
- `Web` MDN Web Docs: IF y Else (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/if...else)
- `Web` MDN Web Docs: While y For (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/while)
- `Web` MDN Web Docs: Primeros pasos en JavaScript (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/A_first_splash)

Contenidos Alura:

- `Artículo` Roadmap para principiantes en JavaScript (https://www.aluracursos.com/blog/roadmap-js)
- `Artículo` Guía de JavaScript: qué es y cómo aprender el lenguaje más popular del mundo (https://www.aluracursos.com/blog/guia-de-javascript)
- `Artículo` Funciones en JavaScript (https://www.aluracursos.com/blog/articulo-funciones-en-javascript-)
- `Artículo` Empezar a programar es con JavaScript (https://www.aluracursos.com/blog/empezar-a-programar-es-con-javascript)
- `Artículo` Comenzando con Front-end (https://www.aluracursos.com/blog/comenzando-con-front-end)
- `Artículo` Organiza tu código Javascript de una manera fácil (https://www.aluracursos.com/blog/organiza-tu-codigo-javascript-de-una-manera-facil)
- `Artículo` Escopo en JavaScript (https://www.aluracursos.com/blog/escopo-en-javascript)
- `Artículo` Conociendo Arrow Functions (https://www.aluracursos.com/blog/conociendo-arrow-functions)
- `Artículo` Trabajando con fechas en JavaScript (https://www.aluracursos.com/blog/trabajando-con-fechas-en-javascript)
- `Artículo` Empezando con fetch en Javascript (https://www.aluracursos.com/blog/empezando-con-fetch-en-javascript)
- `Artículo` JavaScript replace: manipulando Strings y regex (https://www.aluracursos.com/blog/javascript-replace-manipulando-strings-regex)
- `Artículo` This, Getters y Setters en clases de Javascript (https://www.aluracursos.com/blog/this-getters-y-setters-clases-de-javascript)

- `Artículo` Empezando con fetch en Javascript (https://www.aluracursos.com/blog/empezando-con-fetch-en-javascript)
- `Artículo` JavaScript: ¿Cuándo debo usar forEach y map? (https://www.aluracursos.com/blog/javascript-cuando-debo-usar-foreach-y-map)
- `Artículo` ¿Cómo funciona el import y export de JavaScript? (https://www.aluracursos.com/blog/como-funciona-el-import-y-export-de-javascript)
- `Artículo` Comprenda la diferencia entre var, let y const en JavaScript (https://www.aluracursos.com/blog/comprenda-diferencia-entre-var-let-y-const-en-javascript)
- `Artículo` Cómo usar operadores de comparación en Javascript (https://www.aluracursos.com/blog/como-utilizar-operadores-de-comparacion-en-javascript)
- `Artículo` Namespaces: cómo evitar conflictos en código JavaScript (https://www.aluracursos.com/blog/namespaces-como-evitar-conflictos)
- `Artículo` Una guía para importar y exportar módulos con JavaScript (https://www.aluracursos.com/blog/una-guia-para-importar-exportar-modulos-con-java-script)
- `Artículo` Javascript: ¿para qué sirve un array? (https://www.aluracursos.com/blog/javascript-para-que-sirve-un-array)
- `YouTube` ¿Qué es JavaScript? (https://www.youtube.com/watch?v=GJfOSoaXk4s&ab_channel=AluraLatam)
- `YouTube` El mejor lenguaje para empezar (https://www.youtube.com/watch?v=Nrp3c6kNyAw)
- `YouTube` Frameworks de Front End (https://www.youtube.com/watch?v=UNeKzI2WHgQ&ab_channel=AluraLatam)
- `YouTube` Bucles y Bucles Anidados en Javascript (https://youtu.be/FEdqGdtZuwc)
- `YouTube` De la Creación al Borrado: Descubre el Poder del CRUD en el Desarrollo de Aplicaciones (https://youtu.be/jr_98HCSTZc)
- `Curso` JavaScript: primeros pasos con el lenguaje (https://app.aluracursos.com/course/javascript-primeros-pasos-lenguaje)
- `Curso` JavaScript para Front-end (https://aluracursos.com/formacion-javascript-para-front-end)

DOM - Fundamentos:

- El Document Object Model (DOM) es una interfaz de programación para documentos de la web. Representa la página para que los programas puedan cambiar la estructura, el estilo y el contenido del documento. El DOM representa el documento como nosotros y objetos; de esa forma, los lenguajes de programación pueden interactuar con la página.
- Entender cómo funciona el árbol DOM
- Accesando y manipulando elementos HTML y CSS
- Acceder a los padres e hijos de un elemento
- Insertar un nuevo elemento en el árbol
- Quitar un elemento del árbol
- Esperando un evento en un elemento determinado de la página usando

Contenidos

- `Web` W3Schools: JavaScript HTML DOM Methods (https://www.w3schools.com/js/js_htmldom_methods.asp)
- `Web` W3Schools: JavaScript HTML DOM Elements (https://www.w3schools.com/js/js_htmldom_elements.asp)

- Web W3Schools: JavaScript HTML DOM - Changing HTML (https://www.w3schools.com/js/js_htmldom_html.asp)
- Web W3Schools: JavaScript HTML DOM - Changing CSS (https://www.w3schools.com/js/js_htmldom_css.asp)
- Web W3Schools: JavaScript HTML DOM Events (https://www.w3schools.com/js/js_htmldom_events.asp)
- Web W3Schools: JavaScript HTML DOM EventListener (https://www.w3schools.com/js/js_htmldom_eventlistener.asp)
- Web W3Schools: JavaScript HTML DOM Elements (Nodes) (https://www.w3schools.com/js/js_htmldom_nodes.asp)
- Web MDN Web Docs: Introdução ao DOM (https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)
- Web MDN Web Docs: querySelectorAll() (https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelectorAll)
- Web MDN Web Docs: addEventListener() (https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener)
- Artículo An introduction to the JavaScript DOM (https://medium.com/free-code-camp/an-introduction-to-the-javascript-dom-512463dd62ec)
- Artículo The DOM of Javascript (https://javascript.plainenglish.io/the-dom-of-javascript-848506ebf386)
- YouTube DOM JavaScript (https://www.youtube.com/watch?v=8u4Xef5YtFE&ab_channel=Bluuweb)
- YouTube Javascript manipulacion del DOM (https://www.youtube.com/watch?v=iaLiOXXR8eI&list=PLg9145ptuAijiWx4ixGBCZB0kh6YEeNqf&ab_channel=yacklyon)

Contenidos Alura:

- Artículo Qué es DOM? (https://www.aluracursos.com/blog/que-es-dom)
- Curso JS en la Web: Manipulación del DOM con JavaScript (https://app.aluracursos.com/course/js-web-manipulacion-dom-javascript)

Conceptos de SPA:

- Single-page application o SPA es una aplicación web de una sola página o sitio, que interactúa con el usuario reescribiendo dinámicamente la página web actual con nuevos datos del servidor web, en lugar del método por defecto de un navegador web que carga páginas nuevas enteras. El objetivo es conseguir transiciones más rápidas que hagan que el sitio web parezca más una aplicación nativa.
- Entender qué es una SPA
- Establecer rutas a otras páginas
- Conocer los frameworks SPA
- Comunicarse con APIs

Contenidos

- Web MDN Web Docs: SPA (Single-page application) (https://developer.mozilla.org/en-US/docs/Glossary/SPA)
- Web MPAs versus SPAs (https://docs.astro.build/es/concepts/mpa-vs-spa/)
- Artículo Single page application: definición, funcionamiento y utilidad (https://www.ionos.es/digitalguide/paginas-web/creacion-de-paginas-web/single-page-

application/)

- `Artículo` Qué es una web SPA o single page applications (https://www.incentro.com/es-ES/blog/que-es-web-simple-page-applications)
- `Artículo` JavaScript: Vanilla Single Page Applications (SPA) (https://betterprogramming.pub/js-vanilla-script-spa-1b29b43ea475)
- `YouTube` ¿QUÉ es una web SPA? (https://www.youtube.com/watch?v=Fr5QGdJZBVo)

Contenidos Alura:

- `Artículo` ¿Qué es Single Page Application?(SPA) (https://www.aluracursos.com/blog/single-page-application)

React - Componentes:

- React te permite definir componentes como clases o funciones. Los componentes definidos como clases proporcionan más capacidades. Aceptan entradas arbitrarias (llamadas "accesorios") y devuelven elementos React que describen lo que debería aparecer en la pantalla.
- Comprender cómo funcionan los componentes
- Conociendo la biblioteca de componentes con estilo
- Comprender la diferencia entre clase y componentes funcionales

Contenidos

- `Web` Tu primer componente (https://es.react.dev/learn/your-first-component)
- `Web` Creando componentes en nuestra app de React (https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_components)
- `Web` MDN Web Docs: Primeros pasos con React (https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started)
- `Web` Importar y exportar componentes (https://es.react.dev/learn/importing-and-exporting-components)
- `YouTube` Tutorial de React Js - Mi primer componente (https://www.youtube.com/watch?v=0LwhhotHyII)
- `YouTube` Mejores Extensiones de Visual Studio Code (https://www.youtube.com/watch?v=MNaMJ9bhwFI)
- `YouTube` Estilos CSS para los componentes y la aplicación (https://www.youtube.com/watch?v=ckiyt-JJCRk)

Contenidos Alura:

- `Artículo` React: componentes con Styled Components (https://www.aluracursos.com/blog/react-componentes-con-styled-components)
- `Curso` React: ¡Crea aplicaciones web modernas con React! (https://aluracursos.com/formacion-react)

React - Props:

- Props son objetos que se inyecta en los componentes y proporciona algunos datos que se pueden compartir entre otros componentes en un flujo de datos unidireccional desde un elemento principal a un elemento secundario.
- Las Props son de solo lectura.

- accesorios de paso
- manipulación de accesorios

Contenidos

- Web W3Schools: React Props (https://www.w3schools.com/react/react_props.asp)
- Web Pasar props a un componente (https://es.react.dev/learn/passing-props-to-a-component)
- Web MDN Web Docs: Primeros pasos con React - Variables and props (https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#variables_and_props)
- Artículo Propiedades en React.js (Props) (https://medium.com/@diego.coder/propiedades-en-react-js-props-dc1d42c25c28)
- YouTube Props de un componente (https://www.youtube.com/watch?v=l93Usdo9Z4w)
- YouTube Props en React - Primeros passos (https://www.youtube.com/watch?v=lDFoN0Whlhc)

Contenidos Alura:

- Artículo ¿Que és Prop Drilling? (https://www.aluracursos.com/blog/que-es-prop-drilling)

React Hooks - State:

- Controlar el estado de los componentes.
- Manipular variables
- Actualizar el valor de los elementos.

Contenidos

- Artículo Qué son los hooks y cómo utilizarlos en tu proyecto con React. (https://www.paradigmadigital.com/dev/hooks-como-utilizarlos-react/)
- Artículo React Hooks Tutorial (https://developero.io/blog/react-hooks-tutorial-en-espa%C3%B1ol)
- Artículo Conoce los hooks disponibles en React y cómo utilizarlos en tus componentes (https://vabadus.es/blog/otros/conoce-los-hooks-disponibles-en-react-y-como-utilizarlos-en-tus-componentes)
- Artículo Entendiendo los Hooks de React, ¿Cómo usar useState y useEffect en nuestros componentes? (https://desarrollofront.medium.com/entendiendo-los-hooks-de-react-cómo-usar-usestate-y-useeffect-en-nuestros-componentes-611b9e826dfa)
- Artículo Conociendo en profundidad el React Hook useState (https://somospnt.com/blog/289-conociendo-en-profundidad-el-react-hook-usestate-6)
- Artículo Hook para estados complejos useReducer (https://developero.medium.com/react-hooks-usereducer-4d7b68ce22e2)
- YouTube ¿Qué es un Hook? y ejemplo de useState (https://www.youtube.com/watch?v=9bv1lruO8MM)
- YouTube 4 maneras de manejar Areglos de Objetos con useState CRUD (https://www.youtube.com/watch?v=U3lJ7dsDVaE)

Contenidos Alura:

- Artículo React Hooks: ¿Qué son y cómo funcionan? (https://www.aluracursos.com/blog/react-hooks-que-son-y-como-funcionan)
- Curso React: Hooks, contextos y buenas prácticas (https://app.aluracursos.com/course/react-hooks-contextos-buenas-practicas)

- `Curso` React: Ciclo de vida de los componentes (https://app.aluracursos.com/course/react-ciclo-vida-componentes)

Crear una aplicación React:

- Create React App es una forma oficialmente admitida de crear aplicaciones React de una sola página. Ofrece una configuración de construcción moderna sin configuración.
- Estructuración de un nuevo proyecto React
- Crear una aplicación funcional desde cero

Contenidos

- `Web` Empezando con Create React App (https://create-react-app.dev/docs/getting-started)
- `Web` MDN Web Docs: Primeros pasos con Reaccionar (https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#setting_up_your_first_react_app)
- `Artículo` ¿Qué hace realmente Create React App por ti? (https://medium.com/rewrite-tech/what-does-create-react-app-actually-do-for-you-c7a9354acdc4)
- `Artículo` Cómo configurar un nuevo proyecto Create React App (https://medium.com/@dimterion/how-to-set-up-a-new-create-react-app-project-2bfc2992aa16)

Contenidos Alura:

- `Web` Actualización de React y uso de Vite (https://app.aluracursos.com/extra/alura-mais/actualizacion-de-react-y-uso-de-vite-c235)
- `Artículo` React: componentes con Styled Components (https://www.aluracursos.com/blog/react-componentes-con-styled-components)
- `Artículo` React: ¿Biblioteca o Framework? (https://www.aluracursos.com/blog/react-framework-biblioteca)

React Hooks - Effect:

- UseEffect() es un React Hook que le permite manejar los efectos secundarios en sus componentes funcionales de React.
- Ejecutar un componente solo después de renderizar
- Acceso a los accesorios de un elemento
- API de llamadas

Contenidos

- `Web` W3Schools: React Hooks (https://www.w3schools.com/react/react_hooks.asp)
- `Web` W3Schools: React useEffect - Hooks (https://www.w3schools.com/react/react_useeffect.asp)
- `Web` React: Documentación - Introduciendo el Hooks (https://reactjs.org/docs/hooks-intro.html)
- `Web` React: Documentación - Uso del Hook UseEffect (https://reactjs.org/docs/hooks-effect.html)
- `Artículo` React Hooks, useEffect. Añadiendo funcionalidad en el ciclo de vida de nuestro componente (https://desarrollofront.medium.com/entendiendo-los-hooks-de-react-c%C3%B3mo-usar-usestate-y-useeffect-en-nuestros-componentes-611b9e826dfa)
- `YouTube` Para qué sirve UseEffect en React JS (https://www.youtube.com/watch?v=MIlZtKHTYNU)
- `YouTube` UseEffect explicado a fondo (https://www.youtube.com/watch?v=0_D8ruGVp20)

Contenidos Alura:

- `Curso` React: Hooks, contextos y buenas prácticas (https://aluracursos.com/course/react-hooks-contextos-buenas-practicas)

Nivel 2

React Hooks - Memo:

- El React Hook `useMemo` devuelve un valor memorizado. `useMemo` solo volverá a calcular el valor memorizado cuando una de las dependencias haya cambiado.
- Controlar el estado de las variables externas
- Evitar cálculos costosos en cada render

Contenidos

- `Web` Documentación memo (https://es.react.dev/reference/react/memo)
- `Web` W3Schools: React useMemo Hook (https://www.w3schools.com/react/react_usememo.asp)
- `Web` Referencia de la API de los Hooks (https://es.reactjs.org/docs/hooks-reference.html#usememo)
- `Artículo` React Hook para optimizar cálculos useMemo (https://developero.io/blog/react-hook-usememo)
- `YouTube` useCallback y useMemo (https://www.youtube.com/watch?v=THL1OPn72vo)
- `YouTube` Codevolution: React Hooks Tutorial - useMemo Hook (https://www.youtube.com/watch?v=qySZIzZvZOY)

React Hooks - Callback:

- El Hook `useCallback` de React devuelve una función Callback memorizada.
- Aislamiento de funciones intensivas en recursos para que no se ejecuten automáticamente en cada elemento renderizado

Contenidos

- `Web` React: Nueva documentación - useCallback (https://es.react.dev/reference/react/useCallback)
- `Artículo` useCallback, React.js (https://www.linkedin.com/pulse/usecallback-reactjs-junior-javier-duque-valera/?originalSubdomain=es)
- `Artículo` Cómo usar Memo y usar Callback (https://www.developerway.com/posts/how-to-use-memo-use-callback)
- `YouTube` Tutorial useCallback desde cero - React Hooks (https://www.youtube.com/watch?v=6ruYpcCSMOg)
- `YouTube` useCallback y useMemo en React. ¿Los tendrías que usar siempre? (https://www.youtube.com/watch?v=duh3uKn0qnU)

React Hooks - Ref:

- El Hook `useRef` te permite persistir valores entre renderizaciones.
- Almacenar un valor mutable que no provoca una nueva representación cuando se actualiza
- Acceder a un elemento DOM directamente

Contenidos

- `Web` W3Schools: React useRef Hook (https://www.w3schools.com/react/react_useref.asp)

- Web React: Documentation - useRef (https://reactjs.org/docs/hooks-reference.html#useref)
- Artículo Hook para trabajar con referencias useRef (https://developero.medium.com/hook-para-trabajar-con-referencias-useref-53a1edb3ff98)
- Artículo ¿Cómo funciona el hook useRef en React? (https://dev.to/duxtech/como-rayos-funciona-el-hook-useref-en-react-2lah)
- YouTube React UseRef, cómo emplear este hook en React? (https://www.youtube.com/watch?v=4K4yaFGcoFE)

React - Bibliotecas de Design System:

- Un sistema de diseño es una colección de componentes reutilizables, guiados por estándares claros, que se pueden ensamblar para construir cualquier cantidad de aplicaciones.

Contenidos

- Artículo ¿Qué es un Design System? (https://profile.es/blog/que-es-design-system-ejemplo/)
- Artículo Por qué necesitas un sistema de diseño y cómo tus clientes te lo van a agradecer (https://www.plainconcepts.com/es/sistema-diseno-guia/)
- YouTube De la Idea al Sistema de Diseño - Ep. 1 (https://www.youtube.com/watch?v=jRl6l-s7Mzs&t=2s&ab_channel=FranciscoAguileraG.)
- YouTube System design: Como empezar el diseño de un sistema (https://www.youtube.com/watch?v=kZNr1RfHhow&ab_channel=latincoder)

React Developer Tools:

- React Developer Tools se utiliza para inspeccionar los componentes de React, editar accesorios y estados e identificar problemas de rendimiento.
- Depuración de aplicaciones

Contenidos

- Web Extensión React Developer Tools para Google Chrome (https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi)
- Web Herramientas de desarrollo de reacción (https://beta.reactjs.org/learn/react-developer-tools)
- Web Generación de perfiles de componentes con DevTools Profiler (https://reactjs.org/docs/optimizing-performance.html#profiling-components-with-the-devtools-profiler)
- Artículo Depurar aplicaciones React con React DevTools (https://blog.logrocket.com/debug-react-applications-with-the-new-react-devtools/)
- YouTube react.school: React Developer Tools - Installing and Basic Usage (https://www.youtube.com/watch?v=-LfSE9ZxZDI)
- YouTube React Redux Devtools - Cómo configurarlas e introducción (https://www.youtube.com/watch?v=1Bg-D76b24o)

Versiones semánticas de front-end:

- SemVer (Semantic Versioning) es un conjunto simple de reglas y requisitos que dictan cómo se asignan e incrementan los números de versión.
- Organización de las dependencias de un proyecto
- Evitar el "infierno de la dependencia"

Contenidos

- `Web` Documentación: Versionado Semántico 2.0.0 (https://semver.org/lang/es/)
- `Artículo` Versiona mejor tu código: versionamiento semántico y commits convencionales (https://blog.thedojo.mx/2021/12/04/versiona-mejor-tu-codigo-versionamiento-semantico-y-commits-convencionales.html)
- `Artículo` Versionado semántico en npm (https://javascript.plainenglish.io/semantic-versioning-in-npm-54d5b02c3536)
- `Artículo` Qué significan los números de versión en NPM (https://bernardoayala.com/npm-versionado/)
- `YouTube` ¿Cómo actualizar las dependencias (https://www.youtube.com/watch?v=dforsUYxHVE)

CSS-in-JS:

- CSS-in-JS se refiere a un patrón en el que CSS se compone utilizando JavaScript en lugar de definirse en archivos externos.
- Manejar código CSS usando JavaScript
- Usar styled-components
- Conocer Styled-JSX

Contenidos

- `Web` React: Nueva documentación - Aplicar estilos CSS (https://react.dev/reference/react-dom/components/common#applying-css-styles)
- `Web` Microsoft Docs: edición de estilo para marcos CSS-in-JS (https://learn.microsoft.com/es-es/microsoft-edge/devtools-guide-chromium/css/css-in-js)
- `Web` Compatibilidad con CSS-in-JS en DevTools (https://developer.chrome.com/blog/css-in-js/)
- `Web` Compatibilidad con CSS en DevTools (https://learn.microsoft.com/es-mx/microsoft-edge/devtools-guide-chromium/css/css-in-js#css-support-in-devtools)
- `Artículo` El caso de CSS-in-JS (https://octuweb.com/css-in-js/)
- `Artículo` Conceptos básicos de Styled Components (https://dev.to/alfredocu/styled-components-basics-spanish-jdn)
- `YouTube` CSS en JS y Styled Components con Emotion y React (https://www.youtube.com/watch?v=DjVGdUM1dHQ)
- `YouTube` Styled Components en React (https://www.youtube.com/watch?v=f-E1zyTfA3Q)

Contenidos Alura:

- `Curso` React: Utilizando Styled Components (https://aluracursos.com/course/react-styled-components)

Styled Components:

- Styled Components le permiten escribir código CSS real para diseñar sus componentes usando literales de plantilla etiquetados y el poder de CSS.
- Manejo de código CSS en componentes

Contenidos

- `Web` Documentación - Styled Components (https://styled-components.com/docs)
- `Artículo` Styled Components Basics (Spanish) (https://dev.to/alfredocu/styled-components-basics-spanish-jdn)

- `Artículo` Cómo Utilizar Styled Components en React (https://www.escuelafrontend.com/styled-components-en-react)
- `YouTube` Uso de la librería Styled Components (https://www.youtube.com/watch?v=-mS5YEQO2AQ)
- `YouTube` Introducción a Styled-Components con React (https://www.youtube.com/watch?v=Nheqmg2z_dg)

Contenidos Alura:

- `Artículo` React: componentes con Styled Components (https://www.aluracursos.com/blog/react-componentes-con-styled-components)

React Router:

- Manipulación de la navegación entre interfaces y componentes

Contenidos

- `Web` W3Schools: React Router (https://www.w3schools.com/react/react_router.asp)
- `Web` División de código basada en rutas (https://reactjs.org/docs/code-splitting.html#route-based-code-splitting)
- `Web` React Router: Documentación (https://reactrouter.com/en/main/start/overview)
- `Web` Overview React Router (https://beta.reactrouter.com/en/main/start/overview)
- `Artículo` Enrutando Nuestras vistas con React Router (https://4geeks.com/es/lesson/routing-our-views-with-react-router-es)
- `Artículo` Tutorial de React Router versión 6 - Cómo navegar a otros componentes y configurar un enrutador (https://www.freecodecamp.org/espanol/news/tutorial-de-react-router-version-6-como-navegar-a-otros-componentes-y-configurar-un-enrutador/)
- `Artículo` Ejemplo de Implementación de React Router v6 (https://www.frames75.com/2021/ejemplo-react-router.html)
- `YouTube` Cómo usar React Router (https://www.youtube.com/watch?v=afDXVnDnBf4)
- `YouTube` React Router | Nested Routes | Rutas anidadas (https://www.youtube.com/watch?v=NoIg5BJtpVE)
- `YouTube` Rutas con Parámetros consumiendo una API (https://www.youtube.com/watch?v=Q9YCIZMj9-M)

Contenidos Alura:

- `Curso` React Router: Navegación en una SPA (https://aluracursos.com/course/react-router-navegacion-spa)

TypeScript - Fundamentos:

- TypeScript es un lenguaje de programación fuertemente tipado que se basa en JavaScript.
- Comprender en profundidad qué son los tipos y la importancia de la programación tipificada
- Aprender qué es TypeScript, por qué se creó, cómo funciona y su relación con JavaScript
- Conocer las herramientas de TypeScript (integración con el editor de código, verificador estático y compilador)
- Escribir código en TypeScript usando sus herramientas (interfaces, enumeración, decoradores, etc.)

Contenidos

- `Web` W3Schools: Introducción a TypeScript (https://www.w3schools.com/typescript/typescript_intro.php)
- `Artículo` TypeScript, ¿qué es y cómo se diferencia de JavaScript? (https://immune.institute/blog/typescript-que-es-como-se-diferencia-javascript)
- `Artículo` Qué es TypeScript? (https://codigofacilito.com/articulos/typescript)
- `Artículo` ¿Qué es TypeScript y por qué debes aprenderlo? (https://ed.team/blog/que-es-typescript-y-por-que-debes-aprenderlo)
- `Artículo` Introducción a Typescript (https://medium.com/@diego.coder/introducción-a-typescript-7add491e256)
- `Artículo` TypeScript: qué es, diferencias con JavaScript y por qué aprenderlo (https://profile.es/blog/que-es-typescript-vs-javascript/#)
- `YouTube` TypeScript - Tutorial (https://www.youtube.com/watch?v=xtp_DuPxo9Q)

Contenidos Alura:

- `YouTube` Ventajas de trabajar con TypeScript #AluraMás (https://www.youtube.com/watch?v=bFrhU0hbo00&ab_channel=AluraLatam)
- `Curso` TypeScript parte 1: evolucionando tu JavaScript (https://app.aluracursos.com/course/typescript-evolucionando-javascript)
- `Curso` TypeScript parte 2: avanzando en el lenguaje (https://app.aluracursos.com/course/typescript-parte-2-avanzando-lenguaje)
- `Curso` Typescript parte 3: técnicas y buenas prácticas (https://app.aluracursos.com/course/typescript-parte-3-tecnicas-buenas-practicas)

React Testing Library:

- React Testing Library se basa en DOM Testing Library al agregar API para trabajar con componentes React. Es un conjunto de ayudantes que le permiten probar los componentes de React sin depender de los detalles de su implementación.
- Probando los componentes de React

Contenidos

- `Web` Configuración de tu entorno de pruebas (https://es.react.dev/blog/2022/03/08/react-18-upgrade-guide#configuring-your-testing-environment)
- `Web` React Testing Library: Documentación (https://testing-library.com/docs/react-testing-library/intro/)
- `Artículo` React Testing Library, ¿cómo testear componentes con y sin hooks? (https://www.paradigmadigital.com/dev/react-testing-library-como-testear-hooks/)
- `Artículo` Por qué usar Libreria de pruebas (https://lucasbernalte.com/blog/por-que-usar-testing-library-en-lugar-de-enzyme)
- `Artículo` Probando aplicaciones React: Jest y React Testing Library (https://tddreactjs.com/tdd-react/)
- `YouTube` Aprende TDD en REACT Components FACIL 🎯 [Test Driven Development] con Jest & Testing Library (https://www.youtube.com/watch?v=gWS-mOkgXRk)
- `YouTube` Testing en React ¡Aprende dese cero! (https://www.youtube.com/watch?v=KYjjtRgg_H0)
- `YouTube` Cómo hacer tests asíncronos en React Testing Library con waitFor (https://www.youtube.com/watch?v=gz4X7vYzGjA)

Contenidos Alura:

- `Curso` React: Automatizando pruebas en aplicaciones front-end (https://aluracursos.com/course/react-automatizando-tests-aplicaciones-front-end)

Jest:

- Jest es un corredor de pruebas de JavaScript que le permite acceder al DOM a través de jsdom. Proporciona una gran velocidad de iteración combinada con funciones potentes como módulos de simulación y temporizadores para que pueda tener más control sobre cómo se ejecuta el código.
- Componentes de prueba

Contenidos

- `Web` Descripción general de las pruebas (https://reactjs.org/docs/testing.html#gatsby-focus-wrapper)
- `Web` Jest: Documentación (https://jestjs.io/es-ES/docs/getting-started)
- `Artículo` Probando Componentes en React Usando Jest: Lo Básico (https://code.tutsplus.com/es/articles/testing-components-in-react-using-jest-the-basics--cms-28934)
- `Artículo` Cómo testear los componentes de React (https://www.freecodecamp.org/espanol/news/como-probar-los-componentes-de-react-la-guia-completa/)
- `YouTube` Introducción a Unit Testing en React con Jest (https://www.youtube.com/watch?v=n5qbzhZUMsY)
- `YouTube` Testing Components en React con Jest paso a paso (https://www.youtube.com/watch?v=FjJu3hcPSCY)

Cypress:

- Cypress es una herramienta de prueba Front-end que permite configurar, escribir, ejecutar y depurar pruebas.

Contenidos

- `Web` Cypress: Documentación (https://docs.cypress.io/guides/overview/why-cypress)
- `Web` Runebook - Cypress: Documentación (https://runebook.dev/es/docs/cypress/)
- `Artículo` Cypress: ¿Qué es? ¿Por qué y cómo lo usamos en Get on Board? (https://medium.com/get-on-board-dev/cypress-qué-es-por-qué-y-cómo-lo-usamos-en-get-on-board-17688b453fdc)
- `Artículo` Una guía práctica para las pruebas de extremo a extremo con Cypress (https://mx.devoteam.com/expert-view/una-guia-practica-para-las-pruebas-de-extremo-a-extremo-con-cypress/)
- `Artículo` Explorando Cypress: La guía definitiva basada en su documentación oficial (https://www.acontracorrientech.com/explorando-cypress-la-guia-definitiva-basada-en-la-documentacion-oficial/)
- `YouTube` Aprende Testing en Cypress como lo hace un Senior en la vida real 🧪 | FullStack Javascript Bootcamp (https://www.youtube.com/watch?v=HDFNjDKKO6A)
- `YouTube` PRUEBAS de API con CYPRESS IO | Tutorial de CYPRESS | Curso de CYPRESS (https://www.youtube.com/watch?v=IppTzTFY0I8)
- `YouTube` 🤖🚀 Tutorial de Cypress | Primer Test en Cypress io | Cypress Español (https://www.youtube.com/watch?v=puyyzaZulKI)

JavaScript- Callbacks y Promises:

- Una Promesa (Promises) es un proxy de un valor que no necesariamente se conoce cuando se crea la promesa. Esto permite que los métodos asíncronos devuelvan valores como los métodos síncronos- en lugar de devolver inmediatamente el valor final, el método asíncrono devuelve la promesa de proporcionar el valor en algún momento en el futuro.
- Una función de devolución de llamada (Callback) es una función que se pasa a otra función como argumento, que luego se invoca dentro de la función externa para completar algún tipo de rutina o acción.
- Una función asíncrona es una función declarada con la palabra clave asíncrona y la palabra clave espera está permitida dentro de ella. Las palabras clave async y await permiten que el comportamiento asincrónico basado en promesas se escriba en un estilo más claro, lo que evita la necesidad de configurar explícitamente cadenas de promesas.
- Comprender el concepto de programación asíncrona
- Escribir código asíncrono entendiendo el concepto de promesas en JavaScript
- Usar métodos, palabras clave y objetos de JavaScript para manejar promesas como 'Async/Await', '.then()', 'Promise', etc.
- Aprender en qué situaciones necesitas usar la programación asíncrona
- Llamar a las API con 'fetch()'

Contenidos

- `Web` Funciones callbacks (https://lenguajejs.com/javascript/asincronia/callbacks/)
- `Web` ¿Qué son las promesas? (https://lenguajejs.com/javascript/asincronia/promesas/)
- `Web` ¿Qué es la asincronía? (https://lenguajejs.com/javascript/asincronia/que-es/)
- `Web` MDN Web Docs: Promises (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Promise)
- `Web` MDN Web Docs: Función Callback (https://developer.mozilla.org/es/docs/Glossary/Callback_function)
- `Web` MDN Web Docs: Introducción a JavaScript asíncrono (https://developer.mozilla.org/es/docs/Learn/JavaScript/Asynchronous/Introducing)
- `Web` MDN Web Docs: Función async (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/async_function)
- `Artículo` Formas de manejar la asincronía en JavaScript (https://carlosazaustre.es/manejando-la-asincronia-en-javascript)
- `Artículo` Cómo usar la API Fetch de JavaScript para obtener datos (https://www.digitalocean.com/community/tutorials/how-to-use-the-javascript-fetch-api-to-get-data-es)
- `Artículo` Tutorial de Fetch API en JavaScript con ejemplos de JS Fetch, Post y Header (https://www.freecodecamp.org/espanol/news/tutorial-de-fetch-api-en-javascript-con-ejemplos-de-js-fetch-post-y-header/)
- `YouTube` Callbacks vs Promises en JavaScript. ¡Entiende las diferencias y la importancia de cada una (https://www.youtube.com/watch?v=frm0CHyeSbE)
- `YouTube` Carlos Azaustre: ¿Cómo funcionan las Promises y Async/Await en JavaScript? (https://www.youtube.com/watch?v=rKK1q7nFt7M)
- `YouTube` Cómo CONSUMIR una API REST con JAVASCRIPT y Fetch + Promises (https://www.youtube.com/watch?v=FJ-w0tf3d_w)

Contenidos Alura:

- `Curso` JS en la Web: CRUD con JavaScript asíncrono (https://aluracursos.com/course/js-web-crud-javascript-asincrono)

JavaScript - Manejo de errores:

- El manejo de errores se refiere a los procedimientos de respuesta y recuperación de las condiciones de error presentes en una aplicación de software. En otras palabras, es el proceso compuesto por la anticipación, detección y resolución de errores de aplicación, programación o comunicación.
- Conocer y manejar las excepciones más comunes
- Conocer los tipos de errores y en qué situaciones se pueden producir
- Comprender cómo Node.js maneja los errores
- Usar 'try' y 'catch' para el manejo de errores
- Aprender en qué ocasiones y cómo usar `throw`
- Creación de excepciones específicas según las necesidades de su aplicación

Contenidos

- `Web` MDN Web Docs: Flujo de control y manejo de errores (https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Control_flow_and_error_handling)
- `Artículo` Guía Definitiva para el Manejo de Errores en JavaScript (https://kinsta.com/es/blog/errores-en-javascript/)
- `Artículo` Errores personalizados, extendiendo Error (https://es.javascript.info/custom-errors)
- `YouTube` Depurar JavaScript con el Navegador y con Visual Studio Code. (https://www.youtube.com/watch?v=CRXMli2ZkS8)
- `YouTube` Manejar errores en JavaScript con Try/Catch (https://www.youtube.com/watch?v=h9YunSGIBQ8)

Contenidos Alura:

- `Artículo` Depurando aplicaciones en el navegador (https://www.aluracursos.com/blog/articulo-depurando-errores-en-el-navefador-formatado)

Babel - Fundamentos:

- Babel es un compilador de JavaScript. Es una cadena de herramientas que se utiliza principalmente para convertir el código ECMAScript 2015+ en una versión de JavaScript compatible con versiones anteriores en navegadores o entornos actuales y anteriores.

Contenidos

- `Web` Documentación: ¿Qué es Babel? (https://babeljs.io/docs/en/)
- `Artículo` ¿Qué es Babel? (https://www.freecodecamp.org/espanol/news/que-es-babel/)
- `Artículo` Configurar Babel en NodeJS (https://mugan86.medium.com/configurar-babel-en-nodejs-525fd101990b)
- `Artículo` Node.js, NPM, Yarn, NVM, Webpack, Babel... (https://mauriciogc.medium.com/1-node-js-npm-yarn-nvm-webpack-babel-dffbdab392bf)
- `Artículo` Transpilado de Javascript con Webpack y Babel (https://desarrolloweb.com/articulos/transpilado-javascript-webpack.html)
- `YouTube` Babel desde cero + Webpack | tutorial práctico (https://www.youtube.com/watch?v=EU53Lg-DSVM)

Nivel 3

Lottie:

- Lottie es una biblioteca que analiza las animaciones de Adobe After Effects exportadas como json con Bodymovin y las renderiza de forma nativa en dispositivos móviles y en la web.

Contenidos

- `Web` Documentación: Lottie (https://airbnb.io/lottie/#/README)
- `Web` Componentes Lottie (https://lottiereact.com/components/Lottie)
- `Web` Cómo usar animaciones Lottie en una aplicación de React (https://lottiefiles.com/es/blog/trabajando-con-lottie/como-usar-animaciones-lottie-en-una-aplicacion-de-react)
- `Artículo` Lottie: qué es, para qué sirve y cómo utilizarlo en tu web (https://refrescandonegocios.com/lottie/#:~:text=Lottie es una biblioteca creada,Aquí Airbnb lo explican debidamente.)
- `Artículo` ¿Cómo usar Lottie animations & React JS? 🖌️ (https://dev.to/franklin030601/como-usar-lottie-animations-react-js-4afj)
- `Artículo` ¿Cómo dar vida a tu web? ¡Aprende a hacer archivos Lottie en After Effects! (https://trazos.net/como-dar-vida-a-tu-web-aprende-a-hacer-archivos-lottie-en-after-effects/#crear)
- `Artículo` Cómo integrar Lottie View Animations en React Native (https://rootstack.com/es/blog/como-integrar-lottie-view-animations-en-react-native)
- `YouTube` ¿Qué es? ¿Cómo funciona? ¡De after effects a WEB (https://www.youtube.com/watch?v=hLUBXENQSOc)

Framer Motion:

- Framer Motion es una biblioteca de movimiento lista para producción para React de Framer.
- Creando animaciones

Contenidos

- `Web` Documentación: Framer Motion (https://www.framer.com/docs/)
- `Artículo` Framer Motion para transiciones y elementos de React (https://community.listopro.com/framer-motion-para-transiciones-y-elementos-de-react/)
- `Artículo` Tutoriales de Framer Motion: Cree animaciones más avanzadas (https://betterprogramming.pub/framer-motion-tutorials-make-more-advanced-animations-4344b686ea0a)
- `YouTube` Midulive: Framer Motion desde cero (https://www.youtube.com/watch?v=4HnLIAX0EoM)
- `YouTube` Animando la aplicación con Framer Motion (https://www.youtube.com/watch?v=mpa35uufP6A&t=3s)

Service Workers:

- Los Service Workers actúan esencialmente como servidores proxy que se ubican entre las aplicaciones web, el navegador y la red (cuando está disponible). Están destinados, entre otras cosas, a permitir la creación de experiencias fuera de línea efectivas, interceptar solicitudes de red y tomar las medidas adecuadas en función de si la red está disponible y actualizar los activos que residen en el servidor. También permitirán el acceso a notificaciones push y API de sincronización en segundo plano.

Contenidos

- Web MDN Web doc: Usar Service Workers (https://developer.mozilla.org/es/docs/Web/API/Service_Worker_API/Using_Service_Workers)
- Web Descripción general del Service Workers (https://developer.chrome.com/docs/workbox/service-worker-overview/)
- Artículo PWA Series: Service Workers, los básicos de la experiencia offline (https://medium.com/samsung-internet-dev/pwa-series-service-workers-los-b%C3%A1sicos-de-la-experiencia-offline-14592542c738)
- Artículo Almacenamiento en caché del service worker y almacenamiento en caché HTTP (https://web.dev/i18n/es/service-worker-caching-and-http-caching/)
- Artículo Service Workers (https://desarrolloweb.com/articulos/service-workers.html)
- Artículo ¿Sabes qué es un «Service Worker»? (https://ekinbe.com/blog/2018/09/03/sabes-que-es-un-service-worker/)
- Artículo Service workers, servicios web más allá del navegador (https://www.arsys.es/blog/service-worker)
- YouTube Introducción al Service Worker (https://www.youtube.com/watch?v=aUtWHiV3RJg)

React Hooks - Form:

- React Hooks Form es una biblioteca que proporciona validación de formularios.

Contenidos

- Web React Hook Form: Documentación (https://react-hook-form.com/get-started)
- Artículo Introducción a React Hook Form (https://medium.com/nowports-tech/introducci%C3%B3n-a-react-hook-form-b3e725b707c4)
- Artículo Comparación De Librerías De React Forms: Formik Vs React Hook Form (https://apiumhub.com/es/tech-blog-barcelona/comparacion-de-librerias-de-react-forms-formik-vs-react-hook-form/)
- YouTube Crea formularios fácilmente con React Hook Forms (https://www.youtube.com/watch?v=GEfOr56nBsc)
- YouTube Validar formulario de login con react hook form version 7 (https://www.youtube.com/watch?v=Zvp4LD-6ZNQ)

Lodash:

- Lodash es una moderna biblioteca de utilidades de JavaScript que ofrece modularidad, rendimiento y extras. Los métodos modulares de Lodash son excelentes para iterar arreglos, objetos y cadenas; manipular y probar valores; y la creación de funciones compuestas.

Contenidos

- Web Documentación: Lodash (https://lodash.com/)
- Web Lodash, ¿qué se puede hacer con JavaScript puro ahora? (https://javascript.com.es/lodash-vs-vanillajs)
- Artículo ¿Qué son las bibliotecas Lodash? (https://agiliacenter.com/bibliotecas-lodash-que-son-y-en-que-consisten/)
- Artículo 10 funciones importantes de Lodash para desarrolladores de JavaScript (https://geekflare.com/es/lodash-functions-for-javascript-developers/)
- Artículo 15 funciones importantes de Lodash para desarrolladores de JavaScript (https://geekflare.com/es/lodash-functions-for-javascript-developers/)

- YouTube Lodash Ejercicio: 1 Introducción a la Serie de Ejercicios de la Librería JavaScript Lodash (https://www.youtube.com/watch?v=0toucjI4aCk&list=PL2PZw96yQChzC0J2jzFaWW8u0IomHfwA4&index=1&ab_channel=JohnOrtizOrdoñez)

GraphQL:

- GraphQL es un nuevo estándar API que proporciona una alternativa más eficiente, potente y flexible a REST. Fue desarrollado y de código abierto por Facebook y ahora lo mantiene una gran comunidad de empresas e individuos de todo el mundo.
- Comprender cómo se utiliza GraphQL en el desarrollo de API
- Creación de API utilizando bibliotecas y marcos GraphQL

Contenidos

- Web GraphQL: Documentación (https://graphql.org/)
- Web Cómo GraphQL: el tutorial Fullstack para GraphQL (https://www.howtographql.com/)
- Web RedHat: ¿Qué es GraphQL? (https://www.redhat.com/en/topics/api/what-is-graphql)
- Artículo ¿Cómo construir una API GraphQL? (https://blog.back4app.com/es/como-construir-una-api-graphql/)
- Artículo Por qué GraphQL es el futuro de las APIs (http://developinginspanish.com/2019/11/16/por-que-graphql-es-el-futuro-de-las-apis/)
- Artículo GraphQL, el futuro de las API (https://blog.ida.cl/experiencia-de-usuario/graphql-el-futuro-de-las-api/)
- Artículo GraphQL vs REST: Todo lo que Necesitas Saber (https://kinsta.com/es/blog/graphql-vs-rest/)
- YouTube ¿Qué es GraphQL? (https://www.youtube.com/watch?v=RreRD41qlpw)
- YouTube miduver: GraphQL desde de cero (https://www.youtube.com/watch?v=QG-qbmW-wes)

Apollo Client:

- Apollo Client es una biblioteca integral de administración de estado para JavaScript que le permite administrar datos locales y remotos con GraphQL.
- Utilizar Apollo para crear un servidor GraphQL
- Conectar a una API

Contenidos

- Web Documentación: Cliente Apollo (https://www.apollographql.com/docs/react/)
- Artículo Apollo, GraphQL, y cómo Redux me arruga la ropa (https://medium.com/@p4bloch/apollo-graphql-y-cómo-redux-me-arruga-la-ropa-12bd071fda9)
- Artículo Apollo Client, ahora con React Hooks (https://www.apollographql.com/blog/announcement/frontend/apollo-client-now-with-react-hooks/)
- YouTube Cómo usar GraphQL en React ⚛ con Apollo Client (https://www.youtube.com/watch?v=sVFocedf-iU)
- YouTube Creando el frontend con React, React-Router, GraphQl y Apollo Client (https://www.youtube.com/watch?v=xxUfQROMbzc)
- YouTube Crea tu servidor GraphQL con Apollo Server (https://www.youtube.com/watch?v=zDrKmi9ph2Q)
- YouTube APRENDE a usar GRAPHQL en tu APP de REACT con APOLLO CLIENT 🚀 (https://www.youtube.com/watch?v=sVFocedf-iU)

Redux Saga:

- Redux Saga es una biblioteca que tiene como objetivo hacer que los efectos secundarios de la aplicación (es decir, cosas asincrónicas como la obtención de datos y cosas impuras como acceder al caché del navegador) sean más fáciles de administrar, más eficientes de ejecutar, fáciles de probar y mejores en el manejo de fallas.

Contenidos

- `Web` Documentación: Saga Redux (https://redux-saga.js.org/docs/introduction/GettingStarted)
- `Artículo` Qué es React, Redux y cómo se relacionan (https://dev.to/arielmirra/que-es-react-y-react-hooks-y-como-se-relacionan-4e1e)
- `Artículo` Redux Sagas vs Thunk. (https://www.paradigmadigital.com/dev/sagas-vs-thunk/)
- `YouTube` Redux-Saga Explicado | Introduccion | Conceptos (https://www.youtube.com/watch?v=FVP8fpFDarI)
- `YouTube` React + Redux y Redux Saga desde Cero (https://www.youtube.com/watch?v=MsQ6VSwcvPI)
- `YouTube` Cómo configurar Redux Saga en tu proyecto de React.Js (https://www.youtube.com/watch?v=Fzg_gEx4q0Y)

NextJS - Fundamentos:

- Next.js es un marco de desarrollo web de código abierto creado por Vercel que permite aplicaciones web basadas en React con representación del lado del servidor y generación de sitios web estáticos.
- Creación de interfaces web
- Disminución de los tiempos de carga de la página
- Representación de páginas del lado del servidor
- Mejorando el rendimiento en React
- Creación de rutas API con funciones sin servicio

Contenidos

- `Web` Documentación: NextJS (https://nextjs.org/docs)
- `Artículo` Next.js: ¿qué es y por qué debes usarlo? (https://todoxampp.com/next-js-que-es-y-por-que-debes-usarlo/)
- `Artículo` NextJS: ¿el futuro de la web? (https://www.aplyca.com/blog/nextjs-el-futuro-web-que-es-nextjs)
- `Artículo` Crear una aplicación Next.js para chat: Añadir Sendbird UIKit a una aplicación Next.js (https://sendbird.com/es/blog/build-nextjs-app-for-chat-with-sendbird-uikit)
- `Artículo` Cómo habilitar la renderización del lado del servidor para una aplicación React (https://www.digitalocean.com/community/tutorials/react-server-side-rendering-es)
- `YouTube` Aprendiendo NextJS, el framework de React con Server Side Rendering (https://www.youtube.com/watch?v=2jxc8DMzt0I)
- `YouTube` Como cambiar tu proyecto de React Js a Next Js. (https://www.youtube.com/watch?v=ERW30xdsLMs)
- `YouTube` NextJS en 5 minutos (https://www.youtube.com/watch?v=kRV23b1hYzw)

Habilidad Auxiliar: Infraestructura y Back-end

Git y GitHub - Fundamentos:

- Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia.
- GitHub es un servicio de hosting para el desarrollo de software y el control de versiones mediante Git.
- Crear un repositorio
- Clonar un repositorio
- Comprometerse, empujar y tirar hacia y desde el repositorio
- Revertir un commit
- Crear de ramas y Pull requests
- Manejar fusiones y conflictos

Contenidos

- `Web` Git: Libro de Consulta (https://git-scm.com/book/es/v2)
- `Web` GitHub Documentación (https://docs.github.com/es)
- `Web` Github Pages Documentación (https://docs.github.com/es/pages/getting-started-with-github-pages/about-github-pages)
- `Web` W3Schools: Git Tutorial (https://www.w3schools.com/git/default.asp?remote=github)
- `Web` Git School - Visualizing Git (https://git-school.github.io/visualizing-git/)
- `Web` Dangit, Git!?! (https://dangitgit.com/es)
- `Artículo` Git and Github Quickstart Tutorial (https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca)
- `YouTube` ¿Qué es Git y cómo funciona? (https://www.youtube.com/watch?v=jGehuhFhtnE)
- `YouTube` Git y Github | Guia Práctico de Git y Github Desde Cero (https://www.youtube.com/watch?v=HiXLkL42tMU)

Contenidos Alura:

- `Artículo` Git y Github: que son y primeros pasos (https://www.aluracursos.com/blog/git-y-github-que-son-y-primeros-pasos)
- `Artículo` Guía sobre cómo instalar Git en diferentes sistemas operativos (https://www.aluracursos.com/blog/guia-sobre-como-instalar-git-en-diferentes-sistemas-operativos)
- `Artículo` Iniciando un repositorio con Git (https://www.aluracursos.com/blog/iniciando-repositorio-con-git)
- `Artículo` Comenzando con Git: aprendiendo a versionar (https://www.aluracursos.com/blog/comenzando-con-git)
- `Artículo` Creando un repositorio remoto en GitHub (https://www.aluracursos.com/blog/creando-repositorio-remoto-en-github)
- `Artículo` Clonando un repositorio con Git y GitHub (https://www.aluracursos.com/blog/clonando-un-repositorio-remoto)
- `Artículo` Paso a Paso para activar tu proyecto en GitHub Pages. (https://www.aluracursos.com/blog/github-pages)

- `Artículo` Cómo escribir un README increíble en tu Github (https://www.aluracursos.com/blog/como-escribir-un-readme-increible-en-tu-github)
- `Artículo` Buenas practicas en git: evitando errores (https://www.aluracursos.com/blog/como-evitar-errores-en-git)
- `Artículo` GIT: Errores de comandos y directorios (https://www.aluracursos.com/blog/git-errores-de-comandos-y-directorios)
- `Artículo` GIT: errores de commits (https://www.aluracursos.com/blog/git-errores-de-commits)
- `Artículo` GIT: Errores de fusión (https://www.aluracursos.com/blog/errores-de-fusion)
- `Artículo` GIT: Errores con el remoto (https://www.aluracursos.com/blog/errores-con-el-remoto)
- `YouTube` Git y GitHub para Principiantes #AluraMás (https://www.youtube.com/watch?v=-LmFK6skG7s&ab_channel=AluraLatam)
- `YouTube` Git y GitHub: Herramientas Esenciales para el Control de Versiones y Colaboración en Desarrollo (https://youtu.be/dw04N616Abw)
- `YouTube` ¿Puedo subir mi proyecto a Github sin líneas de comando? - Git y Github para principiantes (https://www.youtube.com/watch?v=Yfm16TIpcwk&ab_channel=AluraLatam)
- `YouTube` Git y GitHub: Herramientas Esenciales para el Control de Versiones y Colaboración en Desarrollo (https://www.youtube.com/watch?v=dw04N616Abw)
- `YouTube` ¿Puedo subir mi proyecto a Github sin líneas de comando? - Git y Github para principiantes (https://www.youtube.com/watch?v=Yfm16TIpcwk)
- `Curso` Git y GitHub: repositorio, commit y versiones (https://app.aluracursos.com/course/git-github-repositorio-commit-versiones)
- `Curso` Git y Github: estrategias de ramificación, conflictos y Pull Requests (https://www.aluracursos.com/curso-online-git-github-estrategias-ramificacion-conflictos-pull-requests)

HTTP - Fundamentos:

- HTTP significa Protocolo de transferencia de hipertexto. La comunicación entre las computadoras cliente y los servidores web se realiza mediante el envío de solicitudes HTTP y la recepción de respuestas HTTP.
- Comprender la diferencia entre los verbos HTTP
- Probar solicitudes y verificar los códigos de estado en el navegador
- Aprendiendo a hacer una solicitud HTTP en la línea de comando con WGET
- Descargar una imagen con WGET
- Realización de una POST

Contenidos

- `Web` W3Schools: ¿Qué es HTTP? (https://www.w3schools.com/whatis/whatis_http.asp)
- `Web` MDN Web Docs: Una descripción general de HTTP (https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview)
- `Web` MDN Web Docs: Métodos de solicitud HTTP (https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods)
- `Web` MDN Web Docs: HTTP (https://developer.mozilla.org/es/docs/Web/HTTP)
- `Web` MDN Web Docs: Métodos de petición HTTP (https://developer.mozilla.org/es/docs/Web/HTTP/Methods)
- `Web` MDN Web Docs: Mensajes HTTP (https://developer.mozilla.org/es/docs/Web/HTTP/Messages)

- Web HTTP Cats (https://http.cat/)
- Web HTTP Dogs (https://http.dog/)
- YouTube Peticiones, Métodos Http y Códigos de estado. (https://www.youtube.com/watch?v=gBK-Mfa0Iw8)
- YouTube SSL, TLS, HTTPS, HTTP - Explicado Fácilmente (https://www.youtube.com/watch?v=6HJAWFenYx8&ab_channel=ProfeSang)
- YouTube ⭐ Protocolo HTTP 🖥️ Requests y Responses con: GET, POST, PUT, PATCH y DELETE | Desarrollo web 🌎 (https://www.youtube.com/watch?v=I2MihYAj0Iw)
- YouTube REST y los Verbos de HTTP (https://www.youtube.com/watch?v=OHBHeAPoZ8E)

Contenidos Alura:

- Artículo HTTP: Desmitificando el protocolo Web (https://www.aluracursos.com/blog/http-desmitificando-el-protocolo)
- Artículo ¿Cual es la diferencia entre HTTP y HTTPS? (https://www.aluracursos.com/blog/cual-es-la-diferencia-entre-http-y-https)
- Artículo HTTP: Diferencias entre GET y POST (https://www.aluracursos.com/blog/diferencias-entre-get-y-post)
- Artículo Métodos de petición HTTP (https://www.aluracursos.com/blog/metodos-de-peticion-http)
- Curso HTTP: La base de internet (https://app.aluracursos.com/course/http-base-internet)

JSON:

- JSON significa Notación de objetos de JavaScript. Es un formato de texto para almacenar y transportar datos.
- Crear un objeto
- Transformar un objeto en una cadena
- Transformar una cadena en un objeto
- Manipular un objeto

Contenidos

- Web MDN Web Docs: JSON (https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/JSON)
- Web MDN Web Docs: Trabajando con JSON (https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON)
- YouTube ¿Que és JSON y cómo funciona? (https://www.youtube.com/watch?v=z8qk7T_2sWg&ab_channel=SoyDalto)
- YouTube ⭐ ¿Qué es JSON? ¿Cuál es su SINTAXIS? 🖥️ ¿Cómo crear un archivo JSON? | DESARROLLO WEB 🌎 (https://www.youtube.com/watch?v=RhxOTqFbI5Q&ab_channel=TodoCode)

Contenidos Alura:

- Artículo ¿Que es Json? (https://www.aluracursos.com/blog/que-es-json)
- Artículo ¿JSON y Objeto JavaScript son lo mismo? (https://www.aluracursos.com/blog/json-y-objeto-javascript-son-lo-mismo)
- Artículo Simulando una API REST con json-server (https://www.aluracursos.com/blog/simulando-una-api-rest-con-json-server)

- `Artículo` ¿Qué es JSON Web Token? (https://www.aluracursos.com/blog/que-es-json-web-token)
- `Curso` JS en la Web: CRUD con JavaScript asíncrono (https://www.aluracursos.com/curso-online-js-web-crud-javascript-asincrono)

Línea de Comando - Fundamentos:

- CLI es un programa de línea de comandos que acepta la entrada de texto para ejecutar funciones del sistema operativo.
- Conocer los comandos más importantes

Contenidos

- `Web` W3Schools: What is Command Line Interface (CLI)? (https://www.w3schools.com/whatis/whatis_cli.asp)
- `Web` Uso de argumentos de la línea de comandos para Terminal Windows (https://learn.microsoft.com/es-es/windows/terminal/command-line-arguments?tabs=windows)
- `Artículo` Interfaz de línea de comandos o CLI (https://www.computerweekly.com/es/definicion/Interfaz-de-linea-de-comandos-o-CLI)
- `Artículo` El Manual de Comandos de Linux (https://www.freecodecamp.org/espanol/news/comandos-de-linux/)
- `YouTube` Consola vs Terminal vs Shell vs CLI 💻 ¿Qué es la terminal? (https://www.youtube.com/watch?v=1YxHXBsVNGQ&ab_channel=ProgramadorX)
- `YouTube` Aprende la linea de comandos en un mac - bash scripting (https://www.youtube.com/watch?v=vfwA3pUnVOg&ab_channel=Datademia)
- `YouTube` freeCodeCamp.org: Command Line Crash Course (https://www.youtube.com/watch?v=yz7nYlnXLfE)
- `YouTube` Traversy Media: Command Line Crash Course For Beginners - Terminal Commands (https://www.youtube.com/watch?v=uwAqEzhyjtw)
- `YouTube` Comandos Básicos e Intermedios CMD (https://youtu.be/erKosEQaaFc)
- `YouTube` Terminal MAC tutorial en Español - Cómo usar la terminal en MAC (https://www.youtube.com/watch?v=TmP3y7Z2kk4)

Contenidos Alura:

- `Artículo` CMD: Sugerencias para trabajar en el prompt de Windows (https://www.aluracursos.com/blog/consejos-para-trabajar-en-el-indicador-de-windows)
- `Artículo` Como usar el terminal integrado de Visual Studio Code (https://www.aluracursos.com/blog/como-usar-el-terminal-integrado-de-visual-studio-code)
- `Curso` Linux 1: conociendo y utilizando la terminal (https://app.aluracursos.com/course/linux-1-conociendo-utilizando-terminal)

Cloud - Fundamentos:

- La computación en nube, o cloud computing, es la distribución de servicios informáticos a través de Internet mediante un modelo de tarificación de pago por uso. Una nube se compone de varios recursos informatizados, desde los propios ordenadores (o instancias, en terminología de nube) hasta las redes, el almacenamiento, las bases de datos y todo lo que les rodea. En otras palabras, todo lo que normalmente se necesita para montar el equivalente a una sala de servidores, o incluso un centro de datos completo, estará listo para usar, configurar y ejecutar.
- Conocer la diferencia entre IaaS, PaaS y SaaS
- Conocer los mayores proveedores de nube

- Especializarse en un proveedor específico de su preferencia

Contenidos

- `Web` ¿Qué es la informática en la nube? | Microsoft Azure (https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-cloud-computing/)
- `Web` Amazon AWS: ¿Qué es la computación en nube? (https://aws.amazon.com/en/what-is-cloud-computing/)
- `Web` Tipos de computación en la nube (https://aws.amazon.com/es/types-of-cloud-computing/)
- `Web` ¿Cómo funciona Azure? (https://learn.microsoft.com/es-es/azure/cloud-adoption-framework/get-started/what-is-azure)
- `Web` ¿Qué es el almacenamiento en la nube? (https://aws.amazon.com/es/what-is/cloud-storage/)
- `Web` ¿Qué es la seguridad en la nube? (https://cloud.google.com/learn/what-is-cloud-security?hl=es-419)
- `Web` Arquitectura sin servidor (https://learn.microsoft.com/es-es/dotnet/architecture/serverless/serverless-architecture)
- `Web` ¿Qué es Docker? (https://aws.amazon.com/es/docker/)
- `Artículo` Guía para principiantes sobre los fundamentos de la computación en nube (https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/)
- `Artículo` Cloud Computing para principiantes (https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/)
- `Artículo` ¿Qué es Google Cloud y para qué sirve? (https://www.incentro.com/es-ES/blog/que-es-google-cloud-platform)
- `YouTube` ¿Qué es computación en la nube? | ¿Qué es cloud computing? | Explicado en 4 minutos (https://youtu.be/MCKdahh2lSo)
- `YouTube` 📌 CLOUD COMPUTING ¿Qué es IaaS, PaaS y Saas? | Modelos de Servicio Cloud (https://youtu.be/VR8aXePkQ5M)
- `YouTube` ¿Qué es AWS? (https://www.youtube.com/watch?v=x2vrg7HuM6g)
- `YouTube` ¿ Cómo empiezo con Google Cloud? (Hablemos en Cloud) (https://www.youtube.com/watch?v=OiDWqu0oQfo)
- `YouTube` Introducción a la infraestructura de Google Cloud (https://www.youtube.com/watch?v=209DGQCism4)
- `YouTube` ¿Qué es la Computación en la Nube? | AWS desde cero - Parte 1: Introducción (https://www.youtube.com/watch?v=IciVhWQ8npw)

Contenidos Alura:

- `Artículo` ¿Qué es Cloud y sus principales servicios? (https://www.aluracursos.com/blog/que-es-cloud-y-sus-principales-servicios)
- `Artículo` Conociendo Terraform (https://www.aluracursos.com/blog/conociendo-terraform)
- `Artículo` Empezando con Docker (https://www.aluracursos.com/blog/empezando-con-docker)
- `Artículo` Heroku, Vercel y otras opciones de cloud como plataforma (https://www.aluracursos.com/blog/heroku-vercel-y-otras-opciones-de-cloud-como-plataforma)
- `YouTube` Fundamentos del OCI | Contenidos ONE (https://youtu.be/rEgSc0UqX-g)
- `Curso` Curso Oracle Cloud Infrastructure: implementación de una aplicación en la nube (https://app.aluracursos.com/course/oracle-cloud-infrastructure-implementacion-aplicacion-

nube)

- `Curso` Curso Oracle Cloud Infrastructure: base de datos e infraestructura como código (https://app.aluracursos.com/course/oracle-cloud-infrastructure-base-datos-infraestructura-codigo)
- `Curso` Curso Deploy en Amazon EC2: Alta disponibilidad y escalabilidad de una aplicación (https://app.aluracursos.com/course/deploy-amazon-ec2-alta-disponibilidad-escalabilidad)
- `Curso` Curso Amazon Lightsail: Simplificando la nube (https://app.aluracursos.com/course/amazon-lightsail-simplificando-nube)

YARN:

- Yarn es un administrador de paquetes para su código. Le permite usar y compartir código con otros desarrolladores. El código se comparte a través de algo llamado paquete (a veces denominado módulo). Un paquete contiene todo el código que se comparte, así como un archivo package.json que O describe.
- Administrar paquetes
- Administrar dependencias
- Instalar paquetes sin conexión
- Comandos
- El archivo yarn.lock

Contenidos

- `Web` YARN: Documentación (https://classic.yarnpkg.com/en/docs)
- `Web` Documentación YARN en español (https://runebook.dev/es/docs/yarn/)
- `Artículo` NPM vs Yarn Cheat Sheet (https://shift.infinite.red/npm-vs-yarn-cheat-sheet-8755b092e5cc)
- `Artículo` Node.js, NPM, Yarn, NVM, Webpack, Babel... (https://mauriciogc.medium.com/1-node-js-npm-yarn-nvm-webpack-babel-dffbdab392bf)
- `YouTube` ReactJS: de cero a experto. Instalando yarn (https://www.youtube.com/watch?v=qOBaAcSPYFs)
- `YouTube` Como Instalar Yarn y Crear un Proyecto 2023 (https://www.youtube.com/watch?v=7jbDE22Fcz8)

Contenidos Alura:

- `Artículo` Creando y publicando una biblioteca Javascript en el NPM (https://www.aluracursos.com/blog/creando-y-publicando-una-biblioteca-javascript-en-el-npm)
- `Artículo` NPM vs Yarn (https://www.aluracursos.com/blog/npm-vs-yarn)

Habilidad Auxiliar: UX y Design

Sistemas de Diseño:

- Un sistema de diseño es una colección de componentes reutilizables, guiados por estándares claros, que se pueden ensamblar para crear aplicaciones.
- Creación y mantenimiento de bibliotecas que se consumirán y utilizarán como estándar para construir un proyecto.

Contenidos

- `Artículo` Por qué necesitas un sistema de diseño y cómo tus clientes te lo van a agradecer (https://www.plainconcepts.com/es/sistema-diseno-guia/#:~:text=Un%20sistema%20de%20dise%C3%B1o%20es%20un%20conjunto%20de%20reglas%20y,tra
- `Artículo` Guía paso a paso para crear un sistema de diseño (https://www.ranktracker.com/es/blog/a-step-by-step-guide-to-creating-a-design-system/)
- `Artículo` A comprehensive guide to design systems (https://www.invisionapp.com/inside-design/guide-to-design-systems/)
- `Artículo` Design Tokens en tu Design System (https://gonzalo-vasquez-correa.medium.com/design-tokens-en-tu-design-system-90498eeafd49)
- `YouTube` Sistemas de Diseño (Team Library) (https://www.youtube.com/watch?v=WS67rHFTVzc&ab_channel=jonmircha)
- `YouTube` Crear un design System en Figma (https://www.youtube.com/watch?v=hrr_-3Y8j7M&ab_channel=CreatividadenBlanco)

Figma - Fundamentos:

- Figma es una aplicación web colaborativa para el diseño de interfaces. El conjunto de funciones de Figma se centra en la interfaz de usuario y el diseño de la experiencia del usuario, con énfasis en la colaboración en tiempo real, utilizando una variedad de herramientas de creación de prototipos y editor de gráficos vectoriales.
- Crear diseños de página y componentes

Contenidos

- `Artículo` Figma Basics- Primeros pasos en Figma (https://www.figma.com/community/file/923140611594993345)
- `Artículo` Introducing Figma to React (https://medium.com/figma-design/introducing-figma-to-react-d2d545cba3cc)
- `YouTube` Figma para principiantes (https://www.youtube.com/watch?v=GoNzQHc7-qo)

Contenidos Alura:

- `YouTube` ¿Cómo un desarrollador Front End utiliza el Figma? (https://www.youtube.com/watch?v=UuAX5azcvDQ&ab_channel=AluraLatam)

Sistemas de color:

- Definición de una paleta de colores que tenga sentido para una interfaz dada

Contenidos

- `Artículo` Teoría del color. Guía definitiva para comprenderla (https://graffica.info/teoria-del-color-guia-definitiva/)
- `Artículo` La ciencia de los colores (https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea)
- `Artículo` RGB frente a CMYK: una guía de sistemas de color para diseñadores (https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554)
- `Artículo` Color en el diseño de interfaz de usuario: un marco (práctico) (https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e)
- `Artículo` Guía básica de colores de la interfaz de usuario (https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a)

- YouTube Teoria del Color | Diseño Web (https://www.youtube.com/watch?v=g0JTvvalf8s)
- YouTube ¿Qué es el color? Explicación de la Teoría del color (https://www.youtube.com/watch?v=CFn-wPKxRR4)

Cómo usar fuentes:

- Elegir la fuente más adecuada para un proyecto determinado

Contenidos

- Web Fuentes Web (https://developer.mozilla.org/es/docs/Learn/CSS/Styling_text/Web_fonts)
- Artículo Cómo utilizar cualquier fuente/tipografía en nuestra página web (https://www.imaginanet.com/blog/como-utilizar-cualquier-fuente-tipografia-en-nuestra-pagina-web.html)
- YouTube ¿Cómo Cambiar la Fuente a mi Página Web? (https://www.youtube.com/watch?v=JQ6b-7geFc4)
- YouTube Descarga y utiliza Google Fonts de forma local (https://www.youtube.com/watch?v=cTaFBV1Z0dw)

Contenidos Alura:

- Artículo 6 malas prácticas que perjudican usuarios disléxicos (https://www.aluracursos.com/blog/6-malas-practicas-que-perjudican-usuarios-dislexicos)
