

Front-end

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=tPzq8IufGxE>)
- **YouTube** Aprende HTML en 15 Minutos (<https://www.youtube.com/watch?v=mNbnV3aN3KA>)
- **YouTube** HTML - Ejemplos Prácticos básicos (<https://www.youtube.com/watch?v=NMh94GBcCQ>)

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-html-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 exemplo, transformando uma única column 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 alineamiento
- 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: Guía 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/EB4vWLzfVcl>)
- **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>)

Accesibilidad en Javascript:

- La Accesibilidad Digital es la eliminación de barreras en la Web. El concepto presupone que los sitios y portales sean diseñados de modo que todas las personas puedan percibir, entender, navegar e interactuar de manera efectiva con las páginas.
- Escribir código teniendo en cuenta la accesibilidad.

Contenidos

- **Web** MDN Web Docs: Buenas prácticas de accesibilidad CSS y JavaScript (https://developer.mozilla.org/es/docs/Learn/Accessibility/CSS_and_JavaScript)
- **YouTube** Accesibilidad web: qué es y cómo aplicarla correctamente (<https://youtu.be/OkBlttAqCuo>)
- **Curso** Web.dev: Aprende sobre accesibilidad! (<https://web.dev/learn/accessibility/welcome?hl=es-419>)

Contenidos Alura:

- **Artículo** Inclusión y accesibilidad en América Latina (<https://www.aluracursos.com/blog/inclusion-y-accesibilidad-en-america-latina>)

- **Artículo** Directrices de Accesibilidad web (<https://www.aluracursos.com/blog/directrices-de-accesibilidad-web>)
- **Artículo** Accesibilidad web y el posicionamiento en motores de búsqueda (<https://www.aluracursos.com/blog/accesibilidad-web-y-el-posicionamiento-en-motores-de-busqueda>)
- **Artículo** 6 malas prácticas que perjudican usuarios disléxicos (<https://www.aluracursos.com/blog/6-malas-practicas-que-perjudican-usuarios-dislexicos>)
- **Artículo** Unobtrusive JavaScript (<https://www.aluracursos.com/blog/unobtrusive-javascript>)
- **YouTube** Accesibilidad Web | #Aluramás (<https://youtu.be/ngMOsuZL-XE>)
- **Curso** Formación Accesibilidad Web (<https://app.aluracursos.com/formacion-accesibilidad-web>)

Estrategias de SEO:

- SEO significa optimización para motores de búsqueda y se refiere a las estrategias utilizadas para clasificar un sitio web dentro de los motores de búsqueda como Yahoo, Bing y, por supuesto, el más famoso de todos, Google.
- Elegir palabras clave
- Entender cómo Google clasifica las páginas
- Conocer los factores de clasificación
- Hacer Link Building

Contenidos

- **Web** Guía de inicio de optimización para motores de búsqueda (SEO) (<https://developers.google.com/search/docs/fundamentals/seo-starter-guide?hl=es-419>)
- **Artículo** Guía completa de SEO para 2023 (<https://blog.hubspot.es/marketing/guia-completa-seo>)
- **Artículo** Verdaderas diferencias entre SEO y SEM: descubre cuál es la mejor estrategia (<https://hotmart.com/es/blog/seo-y-sem>)
- **Artículo** ¿Cuáles son los tipos y cómo conseguir posicionamiento en Google? (<https://hotmart.com/es/blog/posicionamiento-en-google>)
- **Artículo** Posicionamiento SEO, Guía en el 2022 para dummies (<https://www.marketerosagencia.com/blog/seo/guia-posicionamiento-seo-dummies/>)
- **Artículo** Auditoría SEO: cómo evaluar el estado SEO de una web (<https://www.marketerosagencia.com/blog/seo/auditoria-seo/>)
- **YouTube** ¿Qué es SEO? (https://youtu.be/_hNCh4oF4IE)
- **YouTube** Cómo posicionar una Web en Google | Tutorial Completo (<https://youtu.be/J8wycys4u7M>)
- **YouTube** Factores de Posicionamiento SEO en 2023 (Tier list SEO) - BIGSEO Podcast #004 con Javier Martínez (<https://youtu.be/tooy6CaTHWA>)
- **YouTube** [6 Técnicas de SEO] para posicionar un vídeo en YouTube con SEO (<https://youtu.be/2daSjQT55m8>)

Contenidos Alura:

- **Artículo** SEO: Herramientas gratuitas para que tu sitio se posicione mejor (<https://www.aluracursos.com/blog/seo-herramientas-gratuitas-para-que-tu-sitio-se-posicione-mejor>)

Diseño Responsivo:

- Ajustar tus páginas al tamaño de pantalla del usuario
- Media queries
- Conocer el concepto de Mobile first

Contenidos

- **Artículo** ¿Qué es el diseño responsivo? (<https://www.ionos.es/digitalguide/paginas-web/diseño-web/que-es-el-diseño-responsivo/#:~:text=El%20dise%C3%B1o%20responsivo%2C%20tambi%C3%A9n%20dise%C3%B1o,genera>)
- **Artículo** Qué es el diseño responsivo y por qué tu web lo necesita? (<https://trazada.com/que-es-el-diseño-responsivo/>)
- **Artículo** Mobile First vs Diseño Responsivo, diferencias y ventajas de cada uno (<https://www.marketerosagencia.com/blog/diseño-web/mobile-first-vs-diseño-responsivo/>)
- **Artículo** Diseño web responsive vs. adaptable ¿Cuál elegir? (<https://es.wix.com/blog/2022/07/diseño-web-responsive-y-adaptable>)
- **YouTube** Qué es el Responsive Web Design - Bien explicado (<https://youtu.be/2zypRlzlHc>)
- **YouTube** Breakpoints Responsive | Qué es responsive design | Eduardo Fierro Pro (https://youtu.be/_ESkNVyXjdY)

Contenidos Alura:

- **Artículo** Puntos de ruptura en el diseño responsivo (<https://www.aluracursos.com/blog/puntos-de-ruptura-en-el-diseño-responsivo>)
- **Curso** Layouts Responsivos: Trabajando con layouts mobile (<https://app.aluracursos.com/course/layouts-responsivos-layouts-mobile>)

Nivel 2

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 asíncrono 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 - Pruebas:

- La prueba de software es el proceso de evaluación y verificación de que un software realmente hace lo que debería hacer. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento.
- Usar pruebas unitarias
- Usar pruebas de integración
- Usar pruebas de comportamiento (behavior)
- Usar mocks

Contenidos

- **Web** Testing (Pruebas) en Javascript (<https://john-florez.gitbook.io/reactjs-book/testing-pruebas-en-javascript>)
- **Artículo** ¿Qué es la prueba unitaria? (<https://www.nimblework.com/es/agile/pruebas-unitarias/>)
- **Artículo** ¿Qué son las pruebas unitarias y cómo llevar una a cabo? (<https://www.yeeply.com/blog/que-son-pruebas-unitarias/>)

- **Artículo** Los 4 pilares de una buena prueba unitaria (<http://codelapps.com/code/4-pilares-de-una-buena-prueba-unitaria/>)
- **Artículo** Angular Unit Testing (<https://medium.com/@vito1986/angular-unit-testing-8a1479079f84>)
- **Artículo** Cómo usar Testing en Angular con Jasmine y Karma (<https://digital55.com/blog/como-usar-testing-angular-jasmine-karma/>)
- **YouTube** Introducción a TDD (<https://youtu.be/eFED4Vljwwk>)
- **YouTube** Cómo hacer Unit Testing en Angular de forma sencilla (<https://youtu.be/0ucTEskLvso>)
- **YouTube** Pruebas unitarias con Angular 12 (<https://youtu.be/tlqfNVvsHc>)
- **YouTube** Introducción a la Automatización de Pruebas de Software (https://youtu.be/R_hh3jAqn8M)
- **YouTube** Tipos y Niveles de Pruebas de Software. (<https://youtu.be/hSxXuRxA9mo>)
- **YouTube** Objetivos de la Automatización de pruebas | ISTQB TAE (https://youtu.be/_nwkhYhbN6Y)

Contenidos Alura:

- **Artículo** Tipos de pruebas: ¿cuáles son las principales y por qué utilizarlas? (<https://www.aluracursos.com/blog/tipos-de-pruebas-cual-utilizar>)

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-navegador-formatado>)

JavaScript - ES6:

- Conocer las diferencias de esta versión de JavaScript

Contenidos

- **Web** Módulos, introducción (<https://es.javascript.info/modules-intro>)
- **Web** Módulos JavaScript (<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Modules>)
- **Artículo** Introducción a ES6 (<https://www.kodetop.com/introduccion-a-es6/>)
- **Artículo** ES6, la nueva versión de ECMAScript (<https://www.syntonize.com/nueva-version-de-ecmascript-es6/>)
- **Artículo** Introducción a la POO en Javascript moderno: las nuevas clases en ES6 (<https://www.etnassoft.com/2016/12/02/introduccion-a-la-poo-en-javascript-moderno-las-nuevas-clases-en-es6/>)
- **YouTube** Javascript Moderno en 13 Minutos | Fundamentos ES6 (https://youtu.be/E8L_2qkvHts)
- **YouTube** Import y Export en Javascript (ES Modules) - Explicación (<https://youtu.be/0t-Le4kdaMg>)

Contenidos Alura:

- **Artículo** ES6 - Desestructuración de objetos (<https://www.aluracursos.com/blog/desestructuracion-de-objetos>)
- **YouTube** Alura: JavaScript avanzado I: ES6, orientação a objetos e padrões de projetos (<https://www.youtube.com/watch?v=T-JVRDnykYg>)
- **Curso** JavaScript: Profundizando en MVC, Proxy y Factory (<https://app.aluracursos.com/course/javascript-profundizando-mvc-proxy-factory>)
- **Curso** JavaScript: Conozca el navegador y los patrones de diseño (<https://app.aluracursos.com/course/javascript-navegador-patrones-diseno>)

JavaScript - Modularización:

- Dividir partes del código en módulos
- Usar import y export

Contenidos

- **Web** Módulos JavaScript (<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Modules>)
- **Artículo** Modularización en Javascript (<https://medium.com/@sebastianpaduano/modularizaci%C3%B3n-en-javascript-538bd6c75fa>)
- **Artículo** Programación Orientada a Objetos: Patrón Módulo y Clases de JS (<https://anamartinezaguilar.medium.com/programaci%C3%B3n-orientada-a-objetos-patr%C3%B3n-m%C3%B3dulo-y-clases-de-js-8fc2dab1a085>)
- **Artículo** Programación Orientada a Objetos en JavaScript - Explicado con ejemplos (<https://www.freecodecamp.org/espanol/news/programacion-orientada-a-objetos-en-javascript-explicado-con-ejemplos/>)
- **YouTube** Organiza el código JavaScript con módulos {ES6} (https://youtu.be/_Fmsf6U4Gtg)
- **YouTube** Javascript: Entendiendo los módulos (https://youtu.be/bHziHEUrA_w)
- **YouTube** Curso Javascript - #07 Módulos, import & export, export default (https://youtu.be/_X7mQyat63Y)

Contenidos Alura:

- **Artículo** ¿Cómo funciona el import y export de JavaScript? (<https://www.aluracursos.com/blog/como-funciona-el-import-y-export-de-javascript>)

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>)

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=IppTzTFYQI8>)
- **YouTube** 🍷 Tutorial de Cypress | Primer Test en Cypress io | Cypress Español (<https://www.youtube.com/watch?v=puyyzaZuIKI>)

Nivel 3

Estructura de Datos:

- En el contexto de los ordenadores, una estructura de datos es una forma específica de almacenar y organizar los datos en la memoria del ordenador para que esos datos puedan ser fácilmente recuperados y utilizados de forma eficiente cuando sea necesario posteriormente.
- Conocer las principales estructuras de datos
- Implementar las principales estructuras de datos

Contenidos

- **Artículo** ¿Qué es una estructura de datos en programación y para qué se utiliza? (<https://blog.soyhenry.com/que-es-una-estructura-de-datos-en-programacion/>)
- **Web** Estructuras de datos - Documentación Python (<https://docs.python.org/es/3/tutorial/datastructures.html>)
- **Artículo** Estructuras de Datos y Algoritmos en Java (https://programacion.net/articulo/estructuras_de_datos_y_algoritmos_en_java_309/4)
- **Artículo** Estructuras de datos con Java: un enfoque práctico (<http://hp.fciencias.unam.mx/~alg/estructurasDeDatos/>)
- **Artículo** Colecciones: ARRAY, SET y DICTIONARY en Swift (<https://www.swiftbeta.com/colecciones-array-set-y-dictionary-en-swift/>)
- **Artículo** Colecciones en Swift y su manejo de memoria (<https://medium.com/@grago/colecciones-en-swift-y-su-manejo-de-memoria-61a03e236dd>)
- **Artículo** Estructuras de datos en .NET con C# (<https://www.genesisrrios.com/es/blog/estructuras-de-datos-en-csharp/>)
- **YouTube** Estructuras de Datos | Primeros Pasos (<https://youtu.be/Df-sgxGzyTg>)
- **YouTube** Qué son las estructuras de datos (<https://youtu.be/oQOWklDr73E?list=PLTd5ehIj0goMTSK7RRAPBF4wP-Nj5DRvT>)
- **YouTube** ¿Qué son y cómo funcionan los árboles? | Ejemplo de implementación (<https://youtu.be/tBaOQeyXYqg>)
- **YouTube** Aprende: Estructura de Datos con Java (https://www.youtube.com/watch?v=_9ScDWpqhFE)

- **YouTube** Introducción - 1 - Estructuras de Datos en C# (https://www.youtube.com/watch?v=rqagJXbauyA&ab_channel=nicosiored)
- **YouTube** Estructuras de datos con Python en 8 minutos: Listas, Tuplas, Conjuntos y Diccionarios (<https://www.youtube.com/watch?v=v25-m1LOUiU>)
- **YouTube** Programación en Python | Colecciones | Diccionarios (<https://www.youtube.com/watch?v=vAy4IM7NLIQ>)

Contenidos Alura:

- **Artículo** Estructura de datos: introducción (<https://www.aluracursos.com/blog/estructura-de-datos-introduccion>)
- **Artículo** Conociendo tuplas en Python (<https://www.aluracursos.com/blog/conociendo-las-tuplas-en-python>)
- **Artículo** Listas en Python: operaciones básicas (<https://www.aluracursos.com/blog/listas-de-python-operaciones-basicas>)
- **Artículo** Estructura de datos: computación práctica con Java (<https://www.aluracursos.com/blog/estructura-de-datos-computacion-practica-con-java>)
- **Artículo** Python: trabajando con diccionarios (<https://www.aluracursos.com/blog/python-trabajando-con-diccionarios>)
- **Artículo** Iterando una lista en Java (<https://www.aluracursos.com/blog/iterando-una-lista-en-java>)
- **Curso** Python: avanzando en el lenguaje (<https://app.aluracursos.com/course/python-avanzando-lenguaje>)
- **Curso** C#: array y tipos genéricos (<https://app.aluracursos.com/course/csharp-array-tipos-genericos>)
- **Curso** Python Collections: listas y tuplas (https://app.aluracursos.com/course/python-collections-listas-tuplas?utm_source=gnarus&utm_medium=timeline)
- **Curso** Curso de Java: trabajar con listas y colecciones de datos (<https://www.aluracursos.com/curso-online-java-trabajar-listas-colecciones-datos>)
- **Curso** Java y java.util: Colecciones, Wrappers y Lambda expressions (<https://www.aluracursos.com/curso-online-java-util-colecciones-wrappers-lambda-expressions>)

Conceptos de Orientación a Objetos:

- La Programación Orientada a Objetos es un paradigma de programación de software basado en la composición e interacción entre diversas unidades llamadas 'objetos' y las clases, que contienen una identidad, propiedades y métodos. Se basa en cuatro componentes de la programación - abstracción digital, encapsulación, herencia y polimorfismo.
- Cómo funcionan los objetos
- Crear y utilizar constructores
- Qué son las clases
- Crear y utilizar métodos
- Cómo funciona la encapsulación
- Qué es la herencia
- Qué es el polimorfismo
- Cómo funcionan las interfaces
- Qué son las abstracciones

Contenidos

- **Web** Programación orientada a objetos (C#) (<https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/tutorials/oop>)
- **Web** Programación orientada a objetos - IBM (<https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming>)
- **Web** Introducción a los objetos JavaScript (<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects>)
- **Artículo** Introducción a POO con JavaScript ES6 (<https://medium.com/academia-hack/introducci%C3%B3n-a-poo-con-javascript-es6-80074fde0cdf>)
- **Artículo** Programación Orientada a Objetos (<https://ellibrodepython.com/programacion-orientada-a-objetos-python>)
- **YouTube** 4 Principios de la Programación Orientada a Objetos (<https://www.youtube.com/watch?v=Uk1C1NARZjU>)
- **YouTube** Mini Curso: POO con PHP (básico) (https://youtu.be/Ben_VC2rm10)
- **YouTube** 🟡 ¿Qué es la Programación Orientada a Objetos en PYTHON? - [Con EJEMPLOS] | Python desde CERO #13 (<https://www.youtube.com/watch?v=KwT1F7uL5rA>)

Contenidos Alura:

- **YouTube** Alura Latam: ¿Qué es la Programación Orientada a Objetos? (<https://www.youtube.com/watch?v=Oigen2sjagk>)
- **Artículo** POO: ¿Qué es la programación orientada a objetos? (<https://www.aluracursos.com/blog/poo-que-es-la-programacion-orientada-a-objetos>)
- **Artículo** Revisando la Orientación a Objetos: encapsulación de Java (<https://www.aluracursos.com/blog/revisando-la-orientacion-a-objetos-encapsulacion-de-java>)
- **Artículo** ¿Qué es encapsulamiento? (<https://www.aluracursos.com/blog/Que-es-encapsulamiento>)
- **Artículo** Herencia en JavaScript (<https://www.aluracursos.com/blog/herencia-en-javascript>)
- **Artículo** Interfaces Gráficas con Eclipse WindowBuilder (<https://www.aluracursos.com/blog/interfaces-graficas-con-eclipse-windowbuilder>)
- **Artículo** Cómo no aprender Java y Orientación a Objetos: getters y setters (<https://www.aluracursos.com/blog/como-no-aprender-java-y-orientacion-a-objetos-getters-y-setters>)
- **Artículo** Ordenar una lista de objetos en Java (<https://www.aluracursos.com/blog/ordenar-una-lista-de-objetos-en-java>)
- **Curso** Formación Java Orientado a Objetos (<https://app.aluracursos.com/formacion-javaoo>)
- **Curso** JavaScript: Introducción a la Orientación a Objetos (<https://app.aluracursos.com/course/javascript-introduccion-orientacion-objetos>)
- **Curso** JavaScript: Herencia e Interfaces en Orientación a Objetos (<https://app.aluracursos.com/course/javascript-herencia-interfaces-orientacion-objetos>)
- **Curso** C# y Orientación a Objetos (<https://app.aluracursos.com/formacion-c-sharpe-orientacion-a-objetos>)
- **Curso** Python: comprensión de la Orientación a Objetos (<https://app.aluracursos.com/course/python-comprension-orientacion-objetos>)
- **Curso** Python: avanzando en la orientación a objetos (<https://app.aluracursos.com/course/python-avanzando-orientacion-objetos>)

- **Curso** Java Polimorfismo: Entendiendo herencia e interfaces (<https://app.aluracursos.com/course/java-parte-3-entendiendo-herencia-interfaces>)
- **Curso** Curso de JavaScript: Objetos (https://app.aluracursos.com/course/javascript-objetos?utm_source=gnarus&utm_medium=timeline)
- **Curso** Formación: Aprenda a programar en Python con orientación a objetos (<https://app.aluracursos.com/formacion-lenguaje-python>)

JavaScript - Almacenamiento:

- Almacenar datos en el front-end con localStorage
- Manipular datos almacenados
- Persistir datos almacenados

Contenidos

- **Web** Almacenamiento del lado cliente (https://developer.mozilla.org/es/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage)
- **Web** Usando la API de almacenamiento web (https://developer.mozilla.org/es/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API)
- **Artículo** Almacenamiento en el navegador con JavaScript y localStorage (<https://parzibyte.me/blog/2019/03/16/almacenamiento-en-navegador-javascript-localstorage/>)
- **Artículo** Almacenamiento local en JavaScript (https://manuais.iessanclemente.net/index.php/Almacenamiento_local_en_JavaScript)
- **Artículo** Almacenamiento local (https://ng-girls.gitbook.io/todo-list-tutorial-spanish/local_storage)
- **Artículo** LocalStorage, sessionStorage (<https://es.javascript.info/localstorage>)
- **Artículo** Persistencia de Datos en Javascript (LocalStorage) (<https://dev.to/maxwellnewage/persistencia-de-datos-en-javascript-localstorage-419m>)
- **YouTube** Qué es LocalStorage y cómo utilizarlo en ejemplo práctico (<https://youtu.be/xzNqwjvHsYg>)
- **YouTube** Guardar y obtener la lista del local storage (<https://youtu.be/XUHYQEJhVzs>)

Contenidos Alura:

- **Curso** JS en la Web: Almacenando datos en el navegador (<https://app.aluracursos.com/course/js-web-almacenando-datos-navegador>)

JavaScript - Concurrencia:

- La programación concurrente es un paradigma de programación para construir programas que utilizan la ejecución simultánea de varias tareas computacionales interactivas, que pueden ser implementadas como programas separados o como un conjunto de hilos creados por un único programa
- Ejecutar tareas en paralelo

Contenidos

- **Web** Hilo principal (https://developer.mozilla.org/es/docs/Glossary/Main_thread)
- **Artículo** Programación multihilos (threads) en Javascript (<https://purojavascript.com/programacion-multihilos-threads-en-javascript/>)
- **Artículo** Multithreading con Javascript: Web Workers. (<https://blog.apside.cl/multithreading-con-javascript-web-workers/>)

- **Artículo** Introducción de Paralelo y Concurrente a la Programación en Python (<https://code.tutsplus.com/es/introduction-to-parallel-and-concurrent-programming-in-python--cms-28612a>)
- **YouTube** Hilos (Threads) en Java (<https://youtu.be/fiLrTdSKVcE>)
- **YouTube** Paralelismo con JavaScript | WebWorkers, multi-hilo. (https://youtu.be/lVd8K_ntKKk)
- **YouTube** Concurrencia vs. Paralelismo (<https://youtu.be/kMr3mF71Kp4>)

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>)

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=RreRD41qIpw>)
- **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=sVFocdf-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=sVFocdf-iU>)

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 | Guía 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=Yfm16Tlpcwk&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=Yfm16Tlpcwk>)
- **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-Mfa0lw8>)
- **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=l2MihYAj0lw>)
- **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](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=RhxOTqFbl5Q&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=yz7nYlnXLF>)
- **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/MCKdahh2ISo>)
- **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=2O9DGQCism4>)
- **YouTube** ¿Qué es la Computación en la Nube? | AWS desde cero - Parte 1: Introducción (<https://www.youtube.com/watch?v=lciVhWQ8npw>)

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 lo 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-dffbda392bf>)
- **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-d2d545c3ba3cc>)
- **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** Teoría 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-ZgeFc4>)
- **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>)

TechGuide - Alura
Alura, PM3 e FIAP
O Techguide.sh é um projeto open source