## OBJETIVOS DE APRENDIZAJES

## **JavaScript CARD VALIDATION** ☐ Manipulación de strings. ☐ Uso de condicionales (if-else | switch | operador HTML y CSS ternario) ☐ Uso de HTML semántico. ☐ Uso de bucles (for | for..in | for..of | while) ☐ Uso de selectores de CSS. ☐ Uso de funciones (parámetros | argumentos | valor de ☐ Construir tu aplicación respetando el diseño realizado (maquetación). ☐ Declaración correcta de variables (const & let) **DOM Testing** ☐ Uso de selectores del DOM. ☐ Testeo unitario. ☐ Manejo de eventos del DOM. ☐ Manipulación dinámica del DOM. (appendChild Estructura del código y guía de estilo |createElement | createTextNode| innerHTML | ☐ Organizar y dividir el código en módulos textContent | etc.) (Modularización) **JavaScript** ☐ Uso de identificadores descriptivos (Nomenclatura | ☐ Manipulación de strings. Semántica) ☐ Uso de condicionales (if-else | switch | operador ☐ Uso de linter (ESLINT) ternario) Git y GitHub ☐ Uso de bucles (for | for..in | for..of | while) ☐ Uso de comandos de git (add | commit | pull | status ☐ Uso de funciones (parámetros | argumentos | valor de ☐ Manejo de repositorios de GitHub (clone | fork | gh-☐ Declaración correcta de variables (const & let) pages) **Testing** UX ☐ Testeo unitario. ☐ Diseñar la aplicación pensando y entendiendo al Estructura del código y guía de estilo ☐ Organizar y dividir el código en módulos ☐ Crear prototipos para obtener feedback e iterar. (Modularización) ☐ Aplicar los principios de diseño visual (contraste, ☐ Uso de identificadores descriptivos (Nomenclatura | alineación, jerarquía) Semántica) ☐ Uso de linter (ESLINT) Git y GitHub **DATA LOVERS** ☐ Uso de comandos de git (add | commit | pull | status HTML y CSS ☐ Manejo de repositorios de GitHub (clone | fork | gh-☐ Uso de HTML semántico. pages) ☐ Uso de selectores de CSS. ☐ Construir tu aplicación respetando el diseño realizado UX (maguetación). ☐ Diseñar la aplicación pensando y entendiendo al ☐ Uso de flexbox en CSS. **DOM y Web APIs** ☐ Crear prototipos para obtener feedback e iterar. ☐ Aplicar los principios de diseño visual (contraste, ☐ Uso de selectores del DOM. alineación, jerarquía) ☐ Manejo de eventos del DOM. ☐ Manipulación dinámica del DOM. (appendChild |createElement | createTextNode| innerHTML | **CIPHER** textContent | etc.) **JavaScript** HTML y CSS ☐ Uso de condicionales (if-else | switch | operador ☐ Uso de HTML semántico. ☐ Uso de selectores de CSS. ☐ Uso de bucles (for | for..in | for..of | while) ☐ Construir tu aplicación respetando el diseño realizado ☐ Uso de funciones (parámetros | argumentos | valor de (maquetación). **DOM** ☐ Manipular arrays (filter | map | sort | reduce) ☐ Uso de selectores del DOM. ☐ Manipular objects (key | value) ☐ Manejo de eventos del DOM. ☐ Uso ES modules (import | export) ☐ Manipulación dinámica del DOM. (appendChild ☐ Diferenciar entre expression y statements. |createElement | createTextNode | innerHTML | ☐ Diferenciar entre tipos de datos atómicos y textContent | etc.) estructurados.

**Testing** 

☐ Testeo unitario.

Estructura del codigo y guia de estilo	Git y Github
☐ Organizar y dividir el código en módulos	☐ Uso de comandos de git (add   commit   pull   status
(Modularización)	push
☐ Uso de identificadores descriptivos (Nomenclatura	☐ Manejo de repositorios de GitHub (clone   fork   gh-
Semántica)	: pages) : □ Colaboración en Github (branches   pull requests
☐ Uso de linter (ESLINT)	tags)
Git y GitHub	☐ Organización en Github (projects   issues   labels
☐ Uso de comandos de git (add   commit   pull   status	milestones)
push)	: Firebase
☐ Manejo de repositorios de GitHub (clone   fork   gh-	☐ Firestore.
pages)	☐ Firebase Auth.
☐ Colaboración en Github (branches   pull requests	☐ Firebase Security rules.
tags)	☐ Observadores. (onAuthStateChanged   onSnapshot)
UX	UX
☐ Diseñar la aplicación pensando y entendiendo al	•
usuario.	□ Diseñar la aplicación pensando y entendiendo al usuario.
☐ Crear prototipos para obtener feedback e iterar.	☐ Crear prototipos para obtener feedback e iterar.
☐ Aplicar los principios de diseño visual (contraste,	☐ Aplicar los principios de diseño visual (contraste,
alineación, jerarquía)	alineación, jerarquía)
☐ Planear y ejecutar tests de usabilidad.	□ Planear y ejecutar tests de usabilidad.
COCIAL NIFTWORK	
SOCIAL NETWORK	BUGER QUEEN
HTML y CSS	HTML y CSS
☐ Uso de HTML semántico.	□ Uso de HTML semántico.
☐ Uso de selectores de CSS.	☐ Uso de selectores de CSS.
☐ Construir tu aplicación respetando el diseño realizado	☐ Construir tu aplicación respetando el diseño realizado
(maquetación). □ Uso de flexbox en CSS.	(maquetación).
DOM y Web APIs	☐ Uso de flexbox en CSS.
☐ Uso de selectores del DOM.	☐ Uso de Media Queries.
☐ Manejo de eventos del DOM.	JavaScript
☐ Manipulación dinámica del DOM. (appendChild	☐ Uso de condicionales (if-else   switch   operador
createElement   createTextNode  innerHTML	ternario)
textContent   etc.)	☐ Uso de funciones (parámetros   argumentos   valor de
☐ History API.	retorno)
□ localStorage.	☐ Manipular arrays (filter   map   sort   reduce) ☐ Manipular objects (key   value)
JavaScript	: ☐ Uso ES modules (import   export)
☐ Uso de condicionales (if-else   switch   operador	☐ Diferenciar entre expression y statements.
ternario)	☐ Diferenciar entre tipos de datos atómicos y
☐ Uso de funciones (parámetros   argumentos   valor de retorno)	estructurados.
☐ Manipular arrays (filter   map   sort   reduce)	☐ Uso de callbacks.
☐ Manipular objects (key   value)	☐ Consumo de Promesas.
☐ Uso ES modules (import   export)	Testing
☐ Diferenciar entre expression y statements.	☐ Testeo unitario.
☐ Diferenciar entre tipos de datos atómicos y	: Estructura del código y guía de estilo
estructurados.	☐ Organizar y dividir el código en módulos
☐ Uso de callbacks.	(Modularización)
☐ Consumo de Promesas.	☐ Uso de identificadores descriptivos (Nomenclatura
Testing	Semántica)
☐ Testeo unitario.	☐ Uso de linter (ESLINT)
☐ Testeo asíncrono.	Git y Github
☐ Uso de librerias de Mock.	☐ Uso de comandos de git (add   commit   pull   status
Estructura del código y guía de estilo	push)
☐ Organizar y dividir el código en módulos	☐ Manejo de repositorios de GitHub (clone   fork   gh-
(Modularización)	pages)
☐ Uso de identificadores descriptivos (Nomenclatura	☐ Colaboración en Github (branches   pull requests
Semántica) □ Uso de linter (ESLINT)	tags)   Organización en Github (projects   issues   labels

Firebase	Testing
<ul><li>☐ Firestore.</li><li>☐ Firebase Auth.</li><li>☐ Firebase security rules.</li><li>☐ Observadores. (onAuthStateChanged   onSnapshot)</li></ul>	☐ Testeo unitario. ☐ Testeo asíncrono. ☐ Uso de librerias de Mock. ☐ Uso de Mocks manuales. ☐ Testeo para múltiples Sistemas Operativos.
Angular	Estructura del código y guía de estilo
<ul> <li>☐ Components &amp; templates.</li> <li>☐ Directivas estructurales (nglf / ngFor)</li> <li>☐ @Input   @Ouput</li> <li>☐ Creación y uso de servicios.</li> <li>☐ Manejos de rutas.</li> <li>☐ Creación y uso Observables.</li> <li>☐ Uso de HttpClient.</li> <li>☐ Estilos de componentes (ngStyle / ngClass)</li> </ul> React	☐ Organizar y dividir el código en módulos (Modularización) ☐ Uso de identificadores descriptivos (Nomenclatura   Semántica) ☐ Uso de linter (ESLINT)  Git y GitHub ☐ Uso de comandos de git (add   commit   pull   status   push)
☐ JSX ☐ Componentes class y componentes function ☐ Props ☐ Eventos en React. ☐ Listas y keys.	<ul> <li>□ Manejo de repositorios de GitHub (clone   fork   ghpages)</li> <li>□ Colaboración en Github (branches   pull requests   tags)</li> <li>□ Organización en Github (projects   issues   labels  </li> </ul>
☐ Renderizado condicional.	milestones)
□ Elevación de estados. □ Hooks	HTTP  ☐ Verbos HTTP (http.get)
☐ CSS modules. ☐ React Router.	Fundamentos de programación
ux	☐ Recursión.
<ul> <li>□ Diseñar la aplicación pensando y entendiendo al usuario.</li> <li>□ Crear prototipos para obtener feedback e iterar.</li> <li>□ Aplicar los principios de diseño visual (contraste, alineación, jerarquía)</li> <li>□ Planear y ejecutar tests de usabilidad.</li> </ul>	
MD-LINKS	
JavaScript	
<ul> <li>Uso de condicionales (if-else   switch   operador ternario)</li> <li>Uso de funciones (parámetros   argumentos   valor de retorno)</li> <li>Manipular arrays (filter   map   sort   reduce)</li> <li>Manipular objects (key   value)</li> <li>Uso ES modules (import   export)</li> <li>Diferenciar entre expression y statements.</li> <li>Diferenciar entre tipos de datos atómicos y estructurados.</li> <li>Uso de callbacks.</li> <li>Consumo de Promesas.</li> <li>Creación de Promesas.</li> </ul>	
Node	
<ul> <li>☐ Uso de sistema de archivos. (fs, path)</li> <li>☐ Instalar y usar módulos. (npm)</li> <li>☐ Creación de modules. (CommonJS)</li> <li>☐ Configuración de package.json.</li> <li>☐ Configuración de npm-scripts</li> <li>☐ Uso de CLI (Command Line Interface - Interfaz de Línea de Comando)</li> </ul>	