

SkyLib: Biblioteca de simulacions sobre conceptes i protocols d'Enginyeria

Isaac Pascual Mauriz

6 de febrer de 2025

Resum– Aquest projecte presenta el desenvolupament de SkyLib, una plataforma web interactiva orientada a l'aprenentatge visual i experimental en enginyeria. Inspirada en PhET i HackTricks, aquesta plataforma ofereix píndoles de coneixement que combinen teoria amb simulacions dinàmiques per facilitar la comprensió de conceptes i protocols. S'ha desenvolupat una arquitectura frontend-only amb hosting estàtic, utilitzant Vue.js, Tailwind.css i HTML5 allotjada a GitHub Pages. A més, s'ha dissenyat una interfície intuïtiva i minimalist per garantir una bona experiència d'usuari. Tot i que la primera píndola sobre el protocol HTTP es troba en fase beta, el projecte estableix una base sòlida per a futures ampliacions, permeten la col·laboració i la incorporació de noves píndoles en el futur.

Paraules clau– Biblioteca, simulació, interactiu, animació, skylib, phet, hacktricks, github pages, vue, tailwind, frontend, backend, open-source, llibreria, col·laboratiu.

Abstract– This project presents the development of SkyLib, an interactive web platform focused on visual and experimental learning in engineering. Inspired by PhET and HackTricks, this platform offers knowledge pills that combine theory with dynamic simulations to facilitate the understanding of concepts and protocols. A frontend-only architecture with static hosting has been developed, using Vue.js, Tailwind CSS, and HTML5, hosted on GitHub Pages. Additionally, an intuitive and minimalist interface has been designed to ensure a seamless user experience. Although the first knowledge pill on the HTTP protocol is currently in beta phase, the project establishes a solid foundation for future expansions, allowing for collaboration and the integration of new knowledge pills in the future.

Keywords– Library, simulation, interactive, animation, skylib, phet, hacktricks, github pages, vue, tailwind, frontend, backend, open-source, bookshop, collaborative.

1 INTRODUCCIÓ

A LGUNA vegada t'has parat a pensar per què, un simple vídeo d'internet pot ser més entenedor que una classe tradicional?

Aquesta diferència es deu, en gran part, al reforç visual que ofereixen aquestes eines. Fenomen que està recolzat per diverses teories de l'aprenentatge, com la Teoria del Processament Dual d'Allan Paivio [10], que afirma que la combinació de canals verbals i visuals millora significativament la comprensió, i la teoria del Constructivisme basada

en les aportacions de Jean Piaget [11] i Lev Vygotsky [12], que destaca la importància de l'aprenentatge actiu a través de l'exploració i l'experiència.

El projecte neix de la meva curiositat per explorar i entendre les complexitats del món que m'envolta. Des de ben petit, he desenvolupat una gran passió per al coneixement en àrees científic-tecnològiques. És d'ençà que el meu pare, professor de biologia i matemàtiques, m'ensanya la pàgina de simulacions PhET¹, que vaig començar a interessar-me per aquest tipus d'eines. Una plataforma gratuïta que, mostra a través de simulacions interactives, com funcionen conceptes complexos de física, química i matemàtiques d'una manera simplificada i visual.

No va ser fins anys més tard, ja a la universitat, que vaig adonar-me que molts conceptes em resultaven molt més fàcils d'entendre i interioritzar quan es presentaven de ma-

- E-mail de contacte: ipasauriz@gmail.com
- Menció realizada: Tecnologies de la Informació
- Treball tutoritzat per: Maria Carmen de Toro Valdivia (Àrea de Ciències de la Computació i Intel·ligència Artificial)
- Curs 2024/25

¹<https://phet.colorado.edu>

nera visual, a través d'esquemes, vídeos explicatius o regles mnemotècniques. És gràcies a això que començó a materialitzar una llavor inicial de la qual acabaria germinant aquest projecte tan ambiciós.

Amb el meu interès recent en el món de la ciberseguretat descobreixo la plataforma HackTricks², una biblioteca digital que permet consultar i aprendre ràpidament conceptes de *hacking*. Un model que em va portar a replantejar la creació d'una plataforma que combines l'estructura organitzada i eficient de HackTricks amb el poder de les simulacions interactives de PhET, amb l'objectiu d'ofrir una experiència d'aprenentatge única i innovadora per als estudiants. Així va sorgir la idea d'aquest projecte, una proposta que no només recopila coneixements tècnics, sinó que també permet als usuaris experimentar i comprendre els conceptes d'una manera més intuïtiva i visual.

La resta del document s'estructura de la següent manera:

Primerament, es presenta el context del treball, on s'exposa la proposta plantejada per al desenvolupament del projecte. Seguidament, a la secció de l'estat de l'art, es contextualitza aquest treball dins el seu marc teòric i es descriuen les tecnologies i eines disponibles. A continuació, es detallen els objectius iniciais del projecte, així com la metodologia emprada en el seu desenvolupament. Posteriorment, l'apartat de planificació descriu les etapes establertes i la seva execució. Aquest apartat és seguit per la secció d'anàlisi i disseny, on es presenten les diferents tecnologiesavalades i l'arquitectura adoptada. Finalment, tenim la secció de desenvolupament, on s'exposen els punts clau del procés d'implementació del projecte.

El document conclou amb la secció de conclusions i línies futures, on s'analitza el grau de compliment dels objectius iniciais i es proposen possibles ampliacions i millores per a treballs futurs. Juntament amb els agrairments, referències i l'apèndix.

2 CONTEXT DEL TREBALL

La proposta que es planteja és la creació d'un entorn web que actui com a biblioteca per a consultar i aprendre ràpidament conceptes, protocols i altres mecanismes que es donen en estudis d'Enginyeria. Conceptes que estaran reforçats per una simulació on l'usuari, a través de la interacció consolidarà millor els coneixements teòrics.

Aquesta biblioteca estarà composta per múltiples píndoles de coneixement, cadascuna dedicada a un tema específic en l'àmbit de l'Enginyeria. Aquestes píndoles cobriran des de conceptes fonamentals, com el protocol HTTP, fins a temes més avançats, com la criptografia RSA o l'ús d'eines de software com Nmap.

Cada píndola combinàrà explicacions teòriques amb representacions visuals i simulacions interactives que permetran als estudiants entendre i consolidar millor els conceptes teòrics.

El projecte està basat en la idea de proporcionar una experiència d'aprenentatge immersiva, on l'usuari pugui experimentar i analitzar de primera mà com funcionen els diferents protocols, algoritmes i eines. A més, un dels objectius principals és aconseguir un alt nivell de precisió en la

representació dels valors, dades i interaccions, amb l'objectiu d'ofrir que aquesta experiència sigui el més pròxim a la realitat possible. D'aquesta manera, no només serviria com a material de suport per a estudiants, sinó que també pot ser incorporada com una eina complementària per al professorat i que s'utilitzi com a ajut en les explicacions teòriques.

Per donar una base sòlida a la dimensió pedagògica d'aquest projecte, s'ha dut a terme una recerca sobre teories i estudis que recolzen la importància de l'aprenentatge més enllà d'un simple input teòric, incorporant també la interacció i el reforç visual com a elements clau. Aquestes teories proporcionen una fonamentació psicològica que justifica i reforça la validesa d'aquest tipus de projectes educatius.

A continuació, profunditzarem en dues teories fonamentals: el constructivisme de Piaget i Vygotsky, que posa èmfasi en l'aprenentatge actiu a través de l'experiència i la teoria del processament dual d'Allan Paivio, que explica com la combinació d'informació verbal i visual pot millorar la retenció del coneixement.

2.1 Constructivisme de Piaget i Vygotsky

El constructivisme és una teoria de l'aprenentatge que defensa que el coneixement no es rep passivament, sinó que es construeix activament a través de l'experiència i la interacció amb l'entorn. Aquest enfocament té dos grans referents: Jean Piaget i Lev Vygotsky, que tot i compartir la idea de l'aprenentatge actiu, van desenvolupar perspectives complementaries sobre com es produeix aquest procés. Basen que es van establir a principis del segle XX entre les dates 1920 i 1950.

L'aprenentatge com a construcció individual de Jean Piaget proposa que l'aprenentatge es desenvolupa a través de l'assimilació i l'acomodació, dos mecanismes a través dels quals l'individu organitza la informació i adapta el seu coneixement a mesura que interactua amb el món. Segons la seva teoria, els infants i els adults no aprenen simplement memoritzant informació, sinó que modifiquen els seus esquemes mentals a mesura que processen nous estímuls i experiències.

En canvi, Lev Vygotsky posa èmfasi en la dimensió social de l'aprenentatge. Segons la seva teoria, el coneixement es desenvolupa mitjançant la interacció amb altres persones, especialment amb figures més experimentades com podrien ser professors, companys, mentors...

Un dels conceptes més rellevants que introduceix és la Zona de Desenvolupament Pròxim (ZDP) [12], que es defineix com la diferència entre el que una persona pot aprendre per si mateix i el que pot adquirir amb ajuda. Això significa que l'aprenentatge es potencia quan hi ha guiatge, ja sigui a través d'un professor, una eina interactiva o un entorn col·laboratiu.

2.2 Teoria del Processament Dual d'Allan Paivio

Desenvolupada per Allan Paivio el 1971, estableix que el cervell humà processa la informació a través de dos sistemes cognitius independents però interconnectats:

- Sistema verbal: Processa la informació provenint del llenguatge i text.

²<https://book.hacktricks.xyz>

- Sistema visual: Processa informació en forma d'imatges, esquemes i representacions gràfiques.

Paivio argumenta que l'aprenentatge és més efectiu quan es combinen simultàniament ambdós sistemes, ja que això permet una codificació i una recuperació de la informació més eficient. Això significa que quan una persona llegeix un text accompanyat d'una representació visual, té més possibilitats d'entendre i memoritzar-lo millor que si simplement rebés la informació de forma independent.

2.3 Aplicació del constructivisme i la teoria del processament dual en aquest projecte

Amb aquest projecte es pretén combinar les dues teories de l'aprenentatge més consolidades per oferir una experiència educativa més efectiva.

D'una banda, el Constructivisme de Piaget i Vygotsky, justifica l'ús de metodologies interactives i experiències pràctiques per ajudar l'usuari a construir activament el seu coneixement. D'altra banda, la Teoria del Processament Dual explica per què la combinació de contingut textual i representacions visuals facilita la comprensió i retenció dels conceptes.

El projecte fusiona ambdues perspectives per donar lloc a una plataforma altament visual i interactiva, on l'aprenentatge es desenvolupa de manera intuïtiva i immersiva. L'enfocament permet que cada píndola de coneixement sigui clara, accessible i de fàcil retenció, oferint a l'usuari una experiència educativa enriquidora. A més, el projecte es basa en una sòlida fonamentació teòrica que reforça la seva validesa com a eina d'aprenentatge eficaç.

3 ESTAT DE L'ART

En aquest apartat s'explica amb detall quines plataformes han servit com a referents per al desenvolupament d'aquest projecte, així com altres recursos interactius existents d'ús lliure. L'objectiu és identificar les solucions actuals en l'àmbit de les simulacions educatives i valorar en quin punt aquesta proposta aporta una innovació dins el camp de l'enginyeria.

A continuació es presenten dues plataformes que han influenciat la concepció del projecte i que comparteixen principis amb la proposta desenvolupada.

La primera és la plataforma PhET, un projecte educatiu desenvolupat l'any 2002 pel premi Nobel Carl Wieman a la Universitat de Colorado Boulder. La iniciativa va sorgir arreu de la seva participació en el projecte Physics 2000 [24], el qual tenia com a objectiu dissenyar simuladors per explicar el comportament físic del condensat Bose-Einstein [25]. Durant aquest procés, Wieman va adonar-se que, els assistents a les seves xerrades, sovint recordaven millor les simulacions que no pas les explicacions teòriques que les accompanyaven. Aquesta observació el va portar a interessar-se en el potencial de les simulacions científiques com a eina d'aprenentatge. Amb els fons obtinguts gràcies al seu Premi Nobel, va fundar PhET, establint així l'inici d'una plataforma dedicada a la creació de simulacions interactives per a facilitar conceptes científics complexos.

A la pàgina web de PhET es poden trobar múltiples simulacions interactives centrades en conceptes de física,

química i matemàtiques entre altres disciplines científiques. Aquestes simulacions permeten als usuaris explorar de manera dinàmica diversos fenòmens, afavorint un aprenentatge progressiu mitjançant l'experiència. L'usuari interactua amb el simulador a través d'una interfície digital que, sense necessitat d'instruccions complexes, li permet entendre el funcionament d'un concepte científic a mesura que el manipula.

Tot i que cada simulació inclou una breu descripció teòrica el seu disseny segueix un enfocament minimalist i intuïtiu, evitant la sobrecàrrega d'informació i reduint l'impacte visual excessiu. El públic objectiu d'aquesta plataforma són principalment estudiants joves, als quals se'ls ofereix una eina per comprendre de manera més pròxima conceptes fonamentals del dia a dia.

La segona referència és HackTricks, una biblioteca d'informació i exemples sobre *hacking*. Aquesta plataforma permet a qualsevol persona aprendre nous conceptes o refrescar coneixements de manera ràpida gràcies a una estructura ben organitzada i un sistema de cerca eficient.

HackTricks va ser creada per Carlos Polop Martín, un enginyer de telecomunicacions espanyol. Polop ha treballat com a *pentester*, *red teamer*, programador i administrador de sistemes. És conegut per les seves contribucions a la comunitat de la ciberseguretat, incloent-hi eines com PEASS i Legion.

Tot i que, el públic objectiu d'aquesta plataforma és més especialitzat que en el cas anterior, HackTricks ofereix una eina valiosa per a professionals i entusiastes de la ciberseguretat que busquen aprofundir en conceptes avançats.

Actualment, existeixen diverses plataformes que ofereixen simulacions independents en un àmbit d'enginyeria. Un exemple destacat és el simulador del protocol TCP, desenvolupat per la Universitat de Berlín [18], que permet visualitzar el funcionament d'aquest protocol.

També es poden trobar altres simulacions allotjades en plataformes independents. Per exemple un simulador de l'algorisme RSA disponible a Cryptool [19], així com una simulació del protocol BGP [20].

No obstant això, aquestes plataformes presenten diverses limitacions. En primer lloc, no hi ha una col·lecció extensa de simulacions dedicades a enginyeria, sinó que es tracta de recursos aïllats sense una estructura homogènia. En segon lloc, la interacció de moltes d'aquestes simulacions és reduïda, limitant-se a representacions esquemàtiques sense possibilitat d'experimentació per part de l'usuari. Aquest factor és clau, ja que, segons la teoria del constructivisme de Piaget i Vygotsky, mencionada a la secció 2.1, l'aprenentatge és més efectiu quan els estudiants poden interactuar amb el material i explorar diferents escenaris en temps real.

Aquestes limitacions són evidenciades en el treball fi de grau de l'Alberto Aguado [17], on es contextualitza l'estat actual dels simuladors TCP. En aquest estudi, l'autor analitza en profunditat les solucions existents, exposa les seves mancances i discuteix les principals limitacions d'aquests simuladors. Les conclusions d'aquest treball no només són rellevants per al camp dels protocols de xarxa, sinó que també poden extrapolar-se a altres àrees abordades en aquest projecte.

Després d'aquesta recerca exhaustiva, puc constatar que no existeix cap eina o plataforma que s'assimili a la pro-

posta desenvolupada en aquest projecte. Aquest buit dins el sector educatiu fa que el projecte sigui una iniciativa innovadora i única, ja que combina l'estructura organitzada i eficient d'una biblioteca de coneixement amb el poder de les simulacions interactives per facilitar la comprensió dels conceptes tècnics complexos. Amb aquesta proposta es busca apropar l'aprenentatge actiu a l'àmbit de les Enginyeries.

4 OBJECTIUS

Aquest projecte té com a objectiu el disseny, desenvolupament i manteniment d'un portal web que ofereix píndoles d'informació de temàtiques variades orientades a Enginyeria. Les píndoles inicials que ens donaran la base del projecte, seran:

- Protocol HTTP sobre TCP/IP [16]: representarem el recorregut complet que realitza una petició HTTP de tipus GET entre un usuari i el servei especificat. Es podrà veure i analitzar com aquest missatge s'encapsula i es transforma entre les diverses capes del model TCP/IP.
- RSA [22]: tindrem una representació visual de com funciona el sistema criptogràfic RSA, des de la creació de les claus públiques i privades fins a l'enviament segur d'un missatge entre dos usuaris per un canal insegur.
- Nmap [21]: veurem com l'eina Nmap funciona a escala forense, depenent del tipus de mode i flag que s'activi en un escaneig d'exploració.

També realitzarem un manual d'usuari perquè aquelles persones que vulguin contribuir al projecte, puguin dissenyar les seves píndoles d'informació seguint un model preestablert. Aquestes propostes, si passen una validació i comprovacions prèvies, seran publicades a la pàgina web. L'objectiu final és crear una comunitat de persones que contribueixin a l'expansió d'aquesta pàgina per abastar la major part dels protocols i mecanismes que tenim avui dia en el món digital.

Finalment, a l'inici del projecte, farem una recerca pedagògica per analitzar i estructurar correctament l'enfocament de l'aprenentatge. Aconseguir desenvolupar píndoles recolzades en les teories del coneixement i pedagògiques presentades en les seccions 2.1 i 2.2, ens brindarà rigor al projecte.

5 METODOLOGIA

En aquest projecte, s'ha adoptat una metodologia híbrida entre la fusió de Scrum [13] i Kanban [14], anomenada Scrumban [15]. Aquesta combinació ens permet estructurar el treball en petites iteracions anomenades sprints, tot mantenint un seguiment visual del progrés amb el taulell Kanban on es mostren les tasques pendents, en curs i finalitzades. A més, utilitzem el sistema de versions GitHub Flow [23], el qual ens permet crear branques individuals per a cada funcionalitat, assegurant que la branca *main* està sempre

estable i amb una versió llesta per desplegar. Aquesta metodologia s'adapta millor a l'enfocament d'aquest projecte i al ritme de treball.

Podem trobar una descripció més detallada de la terminologia i els beneficis de Scrumban i GitHub Flow en l'apèndix A.1.

6 PLANIFICACIÓ

La planificació és una etapa fonamental en el desenvolupament d'aquest projecte, no només facilitarà el seguiment de l'evolució del TFG, sinó que també permetrà identificar possibles desviacions i implementar mesures correctores quan sigui necessari. Tenim cinc grans etapes representades en el diagrama de Gantt (Figura 3 de l'Apèndix) que es divideixen en sprints més curts:

En el primer sprint de l'etapa “estructura i enfoc inicial”, s'ha dut a terme una recerca prèvia sobre les tecnologies, metodologies i informació vinculada als objectius, per obtenir una perspectiva preliminar del projecte i valorar-ne l'abast. A més, s'ha investigat sobre els enfocaments pedagògics actuals que serviran de base per al desenvolupament del projecte.

El següent sprint “disseny web”, ens centrem en el procés d'ideació de la pàgina i el seu contingut, un cop definits clarament els objectius i amb una visió més precisa de la direcció que seguirà el projecte.

Seguidament, es durà a terme l'sprint de “desenvolupament web”, que serà la fase amb més hores de dedicació previstes, ja que implica la construcció de tota la pàgina seguint els dissenys i objectius establerts. Aquesta fase s'ha estructurat de manera que cada component i píndola de coneixement es desenvolupi de forma fragmentada i planificada, garantint un procés més eficient i modular.

També, tenim un sprint de “test i manual d'usuari”, el qual és imprescindible, ja que necessitarem garantir que la pàgina web final passa una sèrie de testos abans de ser publicada. Cal remarcar que en aquest sprint, també entra el disseny del manual d'usuari, adreçat a les persones que vulguin contribuir al projecte i afegir noves píndoles. Aquest manual no es podrà dur a terme que fins que no estigui llest l'apartat de desenvolupament.

Finalment, trobem el sprint de “estudi de vulnerabilitats”. Aquest està plantejat per la realització d'una sèrie de controls i testos que la pàgina haurà de passar per tal de garantir uns estàndards de seguretat. Com que l'estructura de la web serà estàtica, estarà condicionada a altres tipus de vulnerabilitats. En aquest sprint farem un estudi de quines vulnerabilitats pot estar exposada una aplicació web de contingut estàtic i com podem evitar-les.

7 ANÀLISI I DISSENY

En aquest apartat s'exposa el procés d'anàlisi i disseny tècnic del projecte. Es detallen les tecnologies seleccionades i els criteris que han guiat la seva elecció, així com el treball previ al desenvolupament, incloent-hi el disseny estètic i visual de la plataforma.

7.1 Llibreries d'animació

S'han analitzat múltiples opcions disponibles a l'hora d'escol·lir una tecnologia o llibreria per a realitzar les simulacions. L'objectiu era poder realitzar animacions fluides d'objectes i elements del DOM HTML (L'explicació extensa de què és el DOM la trobem a l'apèndix A.3). És essencial per a la simulació que aquestes animacions permetin una interacció directa i en temps real de part de l'usuari. A més de ser tecnologies lleugeres i que fàcilment puguin ser executades en un navegador amb únicament processament frontend, arquitectura que explicarem en la secció 7.2.

En la fase de recerca prèvia vaig seleccionar les següents tecnologies, que analitzarem a continuació detalladament: Fabric.js [4], Konva.js [5], Anime.js [3] i HTML5 [6].

Primerament, trobem Fabric.js i Konva.js, dues llibreries lleugeres de JavaScript que permeten animar i transformar elements en el Canvas d'HTML5. Després d'analitzar ambdues opcions, vaig considerar que Fabric.js era una molt bona alternativa per a la implementació, ja que facilita la gestió d'objectes i la seva manipulació interactiva. De fet, és una llibreria que molt probablement utilitzi en el futur per a altres simulacions.

En segon lloc, tenim Anime.js, una llibreria lleugera de JavaScript per animacions. Aquesta ens proporciona un gran ventall d'animacions i transformacions, molt fluides, que es poden aplicar a objectes del DOM.

Inicialment, el desenvolupament es va iniciar amb Anime.js, ja que semblava una opció molt prometedora per gestionar les animacions de manera estructurada mitjançant *timelines*. No obstant això, a mesura que va avançar la implementació, es va detectar una limitació crítica: una vegada definit el *timeline*, els valors establerts no es podien modificar dinàmicament durant l'execució. Això impedia que l'usuari pogués interactuar en temps real amb l'animació, cosa que va suposar una inversió de temps considerable per intentar trobar una solució.

Després dels inconvenients trobats amb Anime.js, vaig optar per desenvolupar les animacions utilitzant HTML5 natiu, sense dependències externes. Decisió que es va prendre amb l'objectiu de tenir un control total sobre el comportament de les animacions i entendre en profunditat el funcionament del Canvas i la manipulació del DOM. L'ús d'una llibreria com Anime.js o Fabric.js simplifica molts processos, però també pot ocultar detalls importants del funcionament intern d'HTML5 i JavaScript. A més, PhET, utilitza HTML5 sense llibreries addicionals, fet que reforçava la meva elecció.

7.2 Arquitectura de la plataforma

A l'hora de dissenyar un portal web, es poden adoptar diferents tipus d'arquitectures segons les necessitats. Algunes de les opcions més comunes inclouen: pàgines estàtiques, sistemes monolítics, aplicacions amb frontend i backend separats, i estructures content-driven, entre d'altres. Cadascuna d'aquestes presenta avantatges i inconvenients en termes de flexibilitat, escalabilitat i gestió del contingut, per la qual és essencial analitzar quin model s'ajusta millor als objectius del portal.

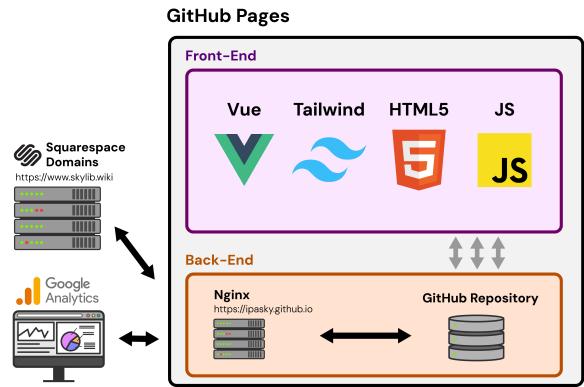


Fig. 1: Diagrama de l'arquitectura de SkyLib.

Una breu descripció dels diferents tipus d'arquitectures que tenim és la següent:

- **Pàgina estàtica:** Aquest model es basa en fitxers HTML, CSS i JavaScript pre-generats, servits directament des del servidor sense cap mena de processament dinàmic. És una arquitectura ràpida, lleugera i fàcil de desplegar, però poc flexible per a continguts que requereixin actualització dinàmica.
- **Arquitectura monolítica:** En aquest model, el frontend, backend i base de dades estan integrats en una única aplicació. És una solució fàcil de gestionar en petits projectes, ja que tota la lògica es troba en un mateix entorn, però pot resultar menys escalable a mesura que l'aplicació creix.
- **Frontend i backend separats:** En aquest altre model, el frontend i el backend es desenvolupen com a sistemes independents, que es comuniquen mitjançant una API (REST o GraphQL). Aquest model aporta més flexibilitat i escalabilitat, especialment en aplicacions modernes.
- **Content-driven:** Basada en la gestió i presentació de continguts estructurats, on el propi contingut és el nucli del sistema.

L'arquitectura del projecte no encaixa de manera estricta en cap d'aquests models, sinó que adopta característiques de diverses arquitectures. Utilitzem un frontend modern desenvolupat amb Vue.js [7], on el contingut és l'element central de l'aplicació, fet que es podria associar al model content-driven. No obstant això, l'aplicació no disposa d'un backend tradicional amb base de dades i API pròpia, sinó que el contingut se serveix de manera estàtica mitjançant GitHub Pages [9]. En aquest sentit, la infraestructura de GitHub Pages utilitza Nginx com a servidor de fitxers estàtics, però sense processament dinàmic.

Així doncs, la terminologia més apropiada per descriure aquest model és "Frontend-only amb hosting estàtic" com podem veure en la figura 1, ja que tenim la lògica de l'aplicació en el frontend i es desplega mitjançant un servei d'allotjament que serveix el codi ja precompilat sense la necessitat d'un backend actiu. Aquest projecte consisteix en una biblioteca de coneixements, no requereix funcionalitats pròpies d'una aplicació dinàmica amb base de dades,

com ara registre d'usuaris, inici de sessió o emmagatzematge d'informació personal. Això fa innecessària la implementació d'un backend, simplificant així l'arquitectura i optimitzant els recursos disponibles.

A més, l'ús de HTML5 per al disseny de les animacions permet crear una experiència interactiva sense dependre de processaments en el servidor, assegurant una càrrega més ràpida i una millor compatibilitat amb diferents dispositius i navegadors. Aquesta arquitectura ens permet oferir una experiència d'usuari fluida i eficient, mantenint la simplicitat i l'accessibilitat com a prioritats fonamentals.

7.2.1 Selecció de Tecnologies per al Frontend

S'ha parlat prèviament que el frontend d'aquest projecte utilitza el framework Vue.js, però és important remarcar els motius d'aquesta elecció i quines altres opcions s'han considerat.

En l'àmbit del frontend, hi ha diverses tecnologies disponibles per desenvolupar aplicacions web modernes. Les més populars actualment són React, Angular i Vue, en aquest ordre, cadascuna amb les seves pròpies característiques.

Tot i que Vue.js no és el framework més popular en comparació amb React o Angular, presenta una sèrie d'avantatges que l'han fet idoni per al projecte:

- Corba d'aprenentatge suau: Vue té una sintaxi clara i accessible, fet que facilita la seva adopció en comparació amb Angular que té una corba d'aprenentatge més pronunciada.
- Estructura modular i flexible: Vue permet una implementació progressiva, fet que facilita la integració de projectes de diferents dimensions.
- Rendiment lleuger i eficient: En comparació amb React i Angular, Vue és més lleuger, cosa que el fa ideal per aplicacions que requereixin una bona optimització de recursos.

No obstant també presenta alguns desavantatges:

- Menys suport: React i Angular tenen suport de Facebook i Google, mentre que Vue es basa més en la seva comunitat de desenvolupadors.
- Menys biblioteques i eines de tercers: Tot i comptar amb un bon ecosistema Vue té una oferta menys extensa que React o Angular, factor que s'ha de considerar de cara a projectes més grans.

Un dels motius principals per triar Vue.js va ser el coneixement previ adquirit durant la carrera. En una de les assig-natures del grau es va treballar amb Vue, fet que va permetre iniciar el desenvolupament sense haver de començar de zero, com hauria estat el cas de React o Angular. Això va representar un estalvi de temps significatiu en la fase inicial i va facilitar la corba d'aprenentatge.

Finalment, dins l'apartat de tecnologies frontend, també s'ha fet ús de Tailwind CSS[8] per al disseny de la plataforma. Aquesta elecció no només ha respost a la necessitat de gestionar el CSS d'una manera més eficient, sinó també al propòsit d'introduir-me en una eina àmpliament utilitzada en grans projectes i entorns professionals.

Optar per Tailwind CSS ha permès familiaritzar-se amb un sistema de disseny escalable i modular, on es gestiona l'estil mitjançant classes unitàries. La implementació amb Tailwind ha suposat una oportunitat per reduir la corba d'aprenentatge en cas d'haver de treballar en el futur amb frameworks similars en projectes més grans.

Pel que fa a les tecnologies disponibles per al backend, aquestes es detallen a l'apèndix A.4, ja que en aquest projecte, no han tingut la mateixa rellevància que les tecnologies del frontend.

7.3 Estètica i disseny artístic

L'estètica d'una pàgina web és un factor determinant a l'hora d'atreure i retenir el públic. Per molt bona que sigui una plataforma web a nivell funcional, si no té un disseny visual atractiu i una bona experiència d'usuari, és probable que no aconsegueixi generar interès. Un disseny desordenat, carregat d'elements innecessaris o amb una interfície confusa pot provocar una experiència frustrant i disminuir la usabilitat de la web.

Aquest apartat se centra en l'anàlisi de l'estètica i el disseny final de la plataforma, així com en els principis d'UX (User Experience) i UI (User Interface) que han tingut un paper fonamental en la seva concepció.

7.3.1 Minimalisme i referents estètics

S'ha optat per un disseny minimalist, on la informació es presenta de manera clara i estructurada, evitant sobrecàrregues visuals que puguin distreure l'usuari. La distribució del contingut es basa en blocs ben definits, que destaquen gràcies a un efecte de vidre semitranslúcida (*glass-morphism*), generant un difuminat suau i agradable a la vista. Aquest estil aporta una sensació de lleugeresa i elegància, facilitant la lectura i navegació dins la plataforma. En la figura 2 es pot veure com s'ha implementat aquest estil minimalist.



Fig. 2: Captura de la portada de SkyLib.wiki

A més, per la naturalesa del nom SkyLib, des d'un principi es va concebre la idea d'implementar un sistema light-dark mode que no només canviés els colors de la interfície, sinó que també modifiqués dinàmicament el fons de la web, generant un efecte més immersiu, com es pot veure en la figura de l'apèndix 5. Per aconseguir aquesta transició fluida entre els diferents modes de lectura, es va decidir crear una animació personalitzada amb Blender [26], una eina open-source molt potent en l'àmbit de la creació 3D.

El procés de desenvolupament d'aquesta animació, va ser un procés d'aprenentatge i experimentació constant, on, a través de diverses sessions de treball, vaig tenir el suport

d'una professional del sector de l'animació i efectes visuals per cinema. Aquest enfocament col·laboratiu va ser fonamental per donar forma a una transició immersiva i alineada amb l'estil visual del projecte.

Aquest projecte ha tingut de grans referents de disseny digital, Apple i Porsche, marques reconegudes per la seva aposta en l'elegància, simplicitat i sofisticació en la comunicació del seu contingut.

Prèviament a la implementació de l'estil del portal, es van realitzar diversos prototips de tipus *wireframe* com es pot veure en la figura 4, per tal de representar ràpidament com havia de ser la web final. Aquest procés ha estat essencial per mantenir una coherència estètica i funcional al llarg del desenvolupament.

7.3.2 UX i UI en el disseny web

A l'hora de dissenyar una pàgina web, és essencial tenir en compte dos conceptes clau:

- UI (User Interface): Fa referència a l'aspecte visual de la web, és a dir, com es veu i com està dissenyada la interfície. Inclou elements com la tipografia, els colors, botons, i distribució de contingut. Un bon disseny UI ha de ser atractiu i alhora mantenir una coherència visual i estètica que faciliti la navegació.
- UX (User Experience): Se centra en com se sent l'usuari quan interactua amb aquesta web, garantint que la seva experiència sigui fluida, intuïtiva i satisfactòria. Això implica optimitzar la navegació, assegurar la càrrega ràpida dels components i evitar que l'usuari es frustri.

En aquest projecte, UX i UI han tingut un paper fonamental. El disseny minimalist ha estat no només per motius estètics, sinó també per garantir una experiència d'usuari neta i sense distraccions, on la informació sigui fàcil de trobar i comprendre. Tenim un exemple visual de com queda estructurada aquesta informació en l'apartat de simulacions dins la pàgina en la figura 7. A més, s'ha buscitat que la interfície sigui intuïtiva, de manera que qualsevol usuari pugui navegar per la plataforma sense necessitat d'un procés d'adaptació.

D'aquesta manera, la combinació d'UX, UI i un enfocament minimalist converteix aquesta plataforma en una eina visualment atractiva i funcional, optimitzada per oferir una experiència d'aprenentatge còmoda i eficient.

Tal com es mostra a la Figura 8, en accedir a la píndola HTTP, l'usuari és rebut amb una pestanya de tutorial. Aquesta secció introduceix la dinàmica de la píndola, facilitant l'aprenentatge i proporcionant un coneixement previ sobre el seu funcionament.

Per a la creació dels diferents elements visuals i estètics de l'UI (User Interface), s'ha utilitzat la suite d'Adobe, detallada a l'apèndix A.2.

7.3.3 Elecció del nom skylib.wiki

El nom d'un projecte és un element clau a l'hora de generar una primera impressió, ja que ha de ser representatiu, fàcil de recordar i amb cert caràcter distintiu. Un bon nom ha d'estar alineat amb la temàtica del projecte, ser prou concís

per facilitar la memorització i, tenir una connotació que reforci la seva identitat.

En aquest cas, el projecte ha estat batejat com SkyLib³, una combinació dels termes Sky (cel) i Lib (biblioteca), fent referència a una "biblioteca del cel". Aquesta elecció en una analogia directa amb les infraestructures en el núvol (cloud computing), ja que molts dels protocols i mecanismes que s'expliquen a la plataforma són d'ús habitual en aquests entorns. D'aquesta manera, el nom SkyLib reflecteix la naturalesa tecnològica del projecte i la seva connexió amb els sistemes digitals actuals.

A més del nom, va ser necessari escollir un domini web adequat i disponible. Entre les opcions considerades, es va valorar el domini .xyz, una extensió utilitzada en projectes tecnològics moderns. Tot i això, aquest domini també ha estat associat en alguns casos a estafes i bots, fet que podria generar dubtes o desconfiança en els usuaris. Per aquest motiu, finalment es va optar pel domini .wiki, ja que fa una referència més clara a la naturalesa del projecte, alineant-se amb el model de la Wikipedia. A més, aquesta extensió té l'avantatge de ser més fàcil de memoritzar i reconèixer en el context d'un recurs educatiu digital.

8 DESENVOLUPAMENT

En les seccions 7.1, 7.2 i 7.3, hem vist i analitzat la tria de tecnologies i arquitectures que componen aquest projecte. Aquest apartat està dedicat al procés de desenvolupament, es detalla com s'han implementat les funcionalitats del sistema i les fases seguides durant la construcció de la plataforma. A més, s'exposen els reptes i problemàtiques trobades en el desenvolupament i les solicions aplicades. Aquest apartat també inclou fragments de codi i explicacions tècniques sobre els principals components implementats així com la integració de les animacions.

8.1 Framework Vue.js

Un cop establerta la justificació sobre l'elecció de Vue.js com a framework per al desenvolupament del frontend, és important aprofundir en com s'ha estructurat la seva implementació i quines característiques específiques s'han utilitzat.

A l'hora de començar un projecte amb Vue, es pot optar per dos models que defineixen la lògica dels components: Options API i Composition API. La diferència principal radica en la seva organització i nivell de flexibilitat que ofereixen.

L'Options API es fa servir en projectes petits, on tenim un desenvolupament més estructurat i guiat, ja que hi ha parts que estan simplificades per a desenvolupadors que comencen. Amb Composition tenim accés al codi nadiu de Vue, cosa que dificulta lleugerament la corba d'aprenentatge, però permet tenir un control i enteniment total de com estem fent servir el framework. Aquest darrer és l'opció que hem triat per desenvolupar el projecte, ja que ens ofereix més control del framework.

A l'hora d'organitzar i definir l'estructura de fitxers d'un projecte, existeixen múltiples models d'arquitectura. En aquest cas s'ha optat per utilitzar l'arquitectura basada en

³<https://www.skylib.wiki>

components (*Component-Based Architecture*), pel fet que és la més adequada per a l'organització de les píndoles de coneixement.

Aquest model està basat en la creació de components reutilitzables i independents, cadascun encapsulat dins del seu propi directori. Aquesta estructura ens permet tenir les diverses píndoles de coneixement en components aïllats i totalment autònoms, facilitant així el manteniment i reutilització d'aquests dins la plataforma.

Teniu tant el flux d'execució que realitza l'aplicatiu [A.5](#) com quina estructura interna tenen els components [A.6](#) a l'àpèndix.

8.2 Gestió de les animacions amb HTML5

Tal com s'ha mencionat anteriorment, la tecnologia utilitzada per al desenvolupament de les animacions és HTML5 natiu. En aquest apartat s'analitzen les funcions bàsiques que s'han implementat per permetre el control i la manipulació d'objectes dins del DOM.

Cada píndola de coneixement es troba encapsulada en un component independent, i la seva lògica en JavaScript es troba dins el bloc de `<script setup>` del propi component. Podem veure com la píndola queda encapsulada visualment en la figura [6](#). Un aspecte clau en aquest context és la manera en què es carrega i s'executa aquest codi, ja que no tot el codi pot executar-se immediatament en carregar el component, especialment quan cal interactuar amb elements del DOM.

Per gestionar correctament aquesta interacció, s'ha fet ús de la funcionalitat `onMounted()`, pròpia de la Composition API de Vue. Aquesta funció permet diferenciar dues fases de l'execució del codi:

- Codi fora de `onMounted()`: S'executa immediatament en carregar el component. Aquí es poden definir variables i funcions que no depenen del DOM.
- Codi dins de `onMounted()`: S'executa només quan el component està totalment muntat, és a dir, quan els elements del DOM ja estan en el navegador. Aquesta característica és essencial per a les píndoles de coneixement, ja que moltes de les animacions han d'accidir a elements HTML directament des de JavaScript.

Per implementar una animació sobre un element, s'ha dissenyat una funció específica que permet gestionar tot el procés mitjançant una única crida, passant com a paràmetre la informació necessària per definir el comportament de l'animació. Aquesta funció anomenada `function input_animation(...)`, ha estat desenvolupada amb la flexibilitat suficient per ser pausada o alterada en qualsevol moment, assegurant així un control total sobre el procés de l'animació.

Els paràmetres d'entrada que accepta la funció són `setedX`, `setedY`, `settedDuration`, `newScale` i `selectedObject`:

- `setedX` i `setedY`: Defineixen la transformació en els eixos de coordenades X i Y, respectivament. Aquests valors indiquen quan es mourà l'objecte en cada direcció. Per exemple, si es vol desplaçar un objecte a (10,0) i després fer-lo tornar a (0,0), primer s'executarà la funció amb `setedX = 10`, i posteriorment amb `setedX = -10` per restablir la posició original.

- `settedDuration`: Determina la durada de l'animació en mil·lisegons, especificant quant de temps ha de trigar l'objecte a completar la seva transformació.
- `newScale`: Controla l'escalat de l'objecte, cosa que permet augmentar o reduir la seva mida durant l'animació.
- `selectedObject`: Indica quin objecte ha de rebre l'animació, identificant-lo mitjançant el seu camp `id`. Aquest paràmetre és essencial per assegurar que la transformació s'apliqui al component desitjat.

El cor del sistema d'animació es basa en un *timeline*, funció que actua com a eix central on anem definint les diferents transformacions i condicions que s'han d'aplicar de manera seqüencial o simultània. Aquesta estructura permet afegir animacions de manera dinàmica i modular assegurant una execució fluida i controlada.

També trobem altres funcions clau per al control d'aquest *timeline* i per a millorar l'experiència visual:

- Pausar el transcurs del temps: Permet aturar temporalment l'animació i reprendre-la quan sigui necessari.
- Redirigir l'atenció de la càmera: Focalitza un objecte concret dins de l'escena, ajustant la perspectiva per donar un èmfasi a un element determinat.
- Transició suau en l'ompliment de text: Introduceix progressivament text dins d'elements de la interfície per millorar la llegibilitat i l'efecte visual.

Tot aquest sistema s'ha pogut desenvolupar gràcies a l'ús de Promises de JavaScript, que permeten gestionar l'execució de les diferents etapes de l'animació de manera controlada. Cada operació dins del *timeline* pot indicar quan ha finalitzat la seva execució abans de donar pas a la següent, o bé executar-se en paral·lel.

Això s'aconsegueix mitjançant funcions asíncrones `async` i l'ús de `await`, que permet sincronitzar l'execució de les diferents fases de l'animació. Quan una funció dins el *timeline* es necessita que s'esperi que la seva predecessora hagi finalitzat, s'utilitza el `await` al seu inici, mentre que si es vol executar en paral·lel, simplement s'afegeix a la seqüència sense aquest modificador.

A l'àpèndix [A.7](#) tens un exemple de codi molt simple verreure com s'implementa el *timeline* i la gestió d'animacions de manera asincrònica.

8.3 Funcionalitats destacades

Dins del conjunt de funcions implementades, n'hi ha algunes de particular importància que cal explicar detalladament. Una d'elles és: `goto_selected(idName, ms, lambda)`, que té com a objectiu centrar la vista en un objecte concret quan l'usuari el selecciona amb el ratolí o quan és activat mitjançant el *timeline* de l'animació.

Per aconseguir aquesta funcionalitat, s'han aplicat transformacions vectorials, ja que cal calcular el vector de diferència entre les coordenades centrals de l'objecte seleccionat i el centre fix de l'animació. A partir d'aquest càcul, es determina la translació necessària i s'aplica la transformació a través de l'atribut CSS `transform`.

Durant el desenvolupament d'aquesta funció, es va identificar un problema en la gestió de l'escala (`scale`). Quan es volia fer un zoom molt elevat, la naturalesa de la fórmula aplicada provocava un creixement convex, és a dir, a mesura que augmentava el valor de `scale`, el zoom es tornava exponencialment més pronunciat.

Per corregir aquest comportament, es va introduir una funció logarítmica que ens permetia atenuar l'efecte d'aquesta convexitat i suavitzar la progressió del zoom. La funció aplicada és la següent:

$$f(k) = 1 + (\log(k \cdot \text{squareWeight}))^p$$

$$\text{squareWeight} = \frac{\text{width}_{\text{wrapper}} \times \text{height}_{\text{wrapper}}}{\text{width}_{\text{objecte}} \times \text{height}_{\text{objecte}}}$$

A on:

- $f(k)$: És el valor corregit que atenua l'escala en valors molt grans.
- k : Per defecte és un 1, ens permet moure la corba logarítmica.
- `squareWeight`: Representa la relació entre la superfície (àrea en píxels quadrats) del contingidor principal de l'animació i la de l'objecte seleccionat.
- p : Exponent ajustable per controlar la intensitat del càlcul logarítmic.

El valor p es calcula a partir de la suma d'un llindar base de 0.7, amb un valor `lambda` que se li passa per paràmetre a la funció. Això ens permet modificar la intensitat del zoom, suavitzant o accentuant la corba logarítmica.

9 CONCLUSIONS

El projecte ha seguit la planificació inicial establerta, assolint la majoria d'objectius clau. S'ha realitzat una recerca i anàlisi prèvia de metodologies pedagògiques, la qual ens han permès estructurar l'enfocament del projecte i dotarlo de rigor acadèmic. Paral·lelament, s'ha desenvolupat una aplicació web amb una interfície minimalistà, estètica i funcional, garantint una experiència d'usuari intuitiva. Addicionalment, la plataforma compta amb un sistema de navegació totalment implementat.

Tot i això, no tots els objectius inicials s'han completat. La píndola de coneixement sobre HTTP es troba en fase beta, i malgrat ser funcional, encara no està llesta per a la seva publicació. Pel que fa a les altres dues píndoles previstes, el seu desenvolupament es va veure aturat abans d'arribar a un estil funcional. Aquest fet es deu principalment a una qüestió de temps, ja que la implementació de la simulació HTTP ha requerit molta més dedicació del que inicialment s'esperava.

A més, el manual d'usuari tampoc s'ha pogut completar. Per aconseguir una documentació clara i útil, és necessari disposar d'una primera píndola plenament desenvolupada, així com garantir que el codi estigui ben estructurat i comentat. Com la simulació HTTP no està finalitzada, la redacció del manual es troba en una fase inicial que serà degudament acabada quan estigui la píndola HTTP enllistada.

De la mateixa manera, no s'ha pogut realitzar el testing ni l'anàlisi de vulnerabilitats, ja que la càrrega de treball derivada del desenvolupament principal ha acaparat la major part del temps disponible. Aquestes proves requereixen una implementació més madura i estable per poder obtenir resultats significatius. Aquest treball quedarà com una tasca pendent de finalització en el futur i s'inclou dins de les línies de desenvolupament futur, que es detallaran en la secció 10.

És important posar la feina feta en context, en el cas de PhET, s'estima que el desenvolupament d'una sola simulació requereix, de mitjana, 160 hores de disseny, 500 hores de desenvolupament i 40 hores de proves. Considerant que un Treball de Fi de Grau disposa d'aproximadament 300 hores, queda clar que completar totes les píndoles planificades hauria estat inviable en aquest termini.

No obstant això, puc afirmar amb total confiança que el projecte ha estat un èxit rotund. Malgrat les limitacions de temps, s'ha aconseguit transmetre la idea i la finalitat del projecte, demostrant la viabilitat i el potencial. S'ha pogut desenvolupar amb totalitat la plataforma per visualitzar i fer les demostracions a les píndoles. A més, s'ha establert una base sòlida que permetrà desenvolupar noves píndoles en el futur, sigui per part meva o per altres persones que vulguin contribuir al projecte.

10 LÍNIES FUTURES

Aquest projecte té un gran potencial de creixement i ofereix múltiples possibilitats d'expansió i millora. Tot i que s'ha establert una base sòlida, encara hi ha diversos aspectes que requereixen desenvolupament per tal d'aconseguir una versió plenament funcional i accessible per a la comunitat.

En primer lloc, és essencial finalitzar la píndola de coneixement HTTP. Això implica polir i revisar la precisió dels continguts per garantir una experiència fluida.

Un altre aspecte fonamental és la finalització del manual d'usuari, que servirà com a guia per a totes aquelles persones interessades a contribuir al projecte. A més, caldrà refinjar els comentaris del codi i millorar la documentació del GitHub, fent que la plataforma sigui més atractiva i fàcil d'entendre per aquells que vulguin col·laborar.

A llarg termini, aquest projecte també pot servir com a base per a futurs Treballs de Fi de Grau, on estudiants interessats puguin aportar noves píndoles de coneixement dels temes que més els hi interessin, ampliant progressivament la biblioteca de simulacions. Aquesta iniciativa permetria que el projecte evolucionés de manera col·laborativa.

Finalment, una de les línies més prometedores és la possibilitat d'extraure la implementació realitzada amb HTML5 per a la creació i gestió d'animacions interactives i empaquetar-la en una llibreria open-source. Això facilitaria la reutilització del sistema en altres projectes.

En resum, aquest projecte compta amb una base sòlida i un gran recorregut de futur per aprofitar les línies de desenvolupament plantejades, garantint la seva evolució i consolidació com a eina educativa innovadora.

AGRAÏMENTS

En primer lloc, vull expressar el meu sincer agraiement a la meva tutora, Maria Carmen de Toro Valdivia, per la seva

ajuda i implicació al llarg de tot el procés de desenvolupament d'aquest projecte. La seva orientació ha estat fonamental per donar-li forma i rigor acadèmic.

També vull agrair a la meva família, especialment als meus pares i al meu germà, pel suport incondicional, tant en els moments de motivació com en aquells de més dificultat.

Finalment, voldria fer una menció especial a la meva amiga, Gisela Torras, per la seva ajuda en el desenvolupament de l'animació del fons de la plataforma.

A tots vosaltres, gràcies.

REFERÈNCIES

- [1] PhET Interactive Simulations, "PhET: Free interactive math and science simulations." *University of Colorado Boulder*. Accedit: 2, Set. 2024. [Online]. Disponible: <https://phet.colorado.edu>
- [2] HackTricks, "HackTricks - A handbook for penetration testing and cybersecurity." *HackTricks*. Accedit: 2, Set. 2024. [Online]. Disponible: <https://book.hacktricks.wiki/en/index.html>
- [3] Anime.js, "Anime.js - JavaScript Animation Engine." *Anime.js*. Accedit: 5, Set. 2024. [Online]. Disponible: <https://animejs.com>
- [4] Fabric.js, "JavaScript Canvas Library." *Fabric.js*. Accedit: 5, Set. 2024. [Online]. Disponible: <https://fabricjs.com>
- [5] Konva.js, "2D Canvas Library for JavaScript." *Konva.js*. Accedit: 5, Set. 2024. [Online]. Disponible: <https://konvajs.org>
- [6] I. Hickson, "HTML5: A vocabulary and associated APIs for HTML and XHTML." *World Wide Web Consortium (W3C), W3C Recommendation*. Accedit: 7, Set. 2024. [Online]. Disponible: <https://www.w3.org/TR/html5/>
- [7] Vue.js, "The Progressive JavaScript Framework." *Vue*. Accedit: 7, Set. 2024. [Online]. Disponible: <https://vuejs.org>
- [8] Tailwind CSS, "Rapidly build modern websites without ever leaving your HTML." *Tailwind CSS*. Accedit: 8, Set. 2024. [Online]. Disponible: <https://tailwindcss.com>
- [9] GitHub, "GitHub Pages - Websites for you and your projects." *GitHub*. Accedit: 17, Set. 2024. [Online]. Disponible: <https://pages.github.com>
- [10] J. M. Clark and A. Paivio, "Dual coding theory and education," *Educational Psychology Review*, vol. 3, no. 3, pp. 149–210, 1991. Accedit: 19, Set. 2024. [Online]. Disponible: <https://doi.org/10.1007/BF01320076>
- [11] J. Piaget, *La construction du réel chez l'enfant*. Neuchâtel, Suïssa: Delachaux et Niestlé, 1937.
- [12] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press, 1978.
- [13] Atlassian, "¿Qué es Scrum?" *Atlassian*. Accedit: 19, Set. 2024. [Online]. Disponible: <https://www.atlassian.com/es/agile/scrum>
- [14] Atlassian, "¿Qué es Kanban?" *Atlassian*. Accedit: 19, Set. 2024. [Online]. Disponible: <https://www.atlassian.com/es/agile/kanban>
- [15] Atlassian, "¿Qué es Scrumban?" *Atlassian*. Accedit: 19, Set. 2024. [Online]. Disponible: <https://www.atlassian.com/es/agile/project-management/scrumban>
- [16] D. E. Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architectures*, 6th ed. Upper Saddle River, NJ: Pearson, 2014.
- [17] A. Aguado, "Informe final." *Treball de Fi de Grau, Universitat Autònoma de Barcelona*, 2022. [Online]. Disponible: https://ddd.uab.cat/pub/tfg/2022/tfg_1636011/1457903_-Informe_final-ammended-AlbertoAguado.pdf
- [18] Technische Universität Berlin, "Animations for Computer Networks." *Telecommunication Networks Group (TKN), TU Berlin*. Accedit: 21, Set. 2024. [Online]. Disponible: <https://www.tkn.tu-berlin.de/teaching/rn/animations/>
- [19] CrypTool, "RSA Visual - Understanding RSA encryption." *CrypTool Legacy*. Accedit: 21, Set. 2024. [Online]. Disponible: <https://legacy.cryptool.org/en/cto rsa-visual>
- [20] BGPsim, "BGPsim - A BGP simulation tool." *BGPsim Legacy*. Accedit: 21, Set. 2024. [Online]. Disponible: https://bgpsim.github.io/legacy/v0_13/?s=hibernia
- [21] Nmap, "Nmap: Network Mapper - Free Security Scanner." *Nmap*. Accedit: 29, Set. 2024. [Online]. Disponible: <https://nmap.org>
- [22] GeeksforGeeks, "RSA Algorithm in Cryptography." *GeeksforGeeks*. Accedit: 29, Set. 2024. [Online]. Disponible: <https://www.geeksforgeeks.org/rsa-algorithm-cryptography/>
- [23] GitHub Docs, "GitHub Flow." *GitHub Documentation*. Accedit: 20, Set. 2024. [Online]. Disponible: <https://docs.github.com/en/get-started/using-github/github-flow>
- [24] Physics 2000, "Physics 2000 - Interactive Physics Applets." *Physics 2000*. Accedit: 29, Set. 2024. [Online]. Disponible: <https://www.physics2000.com/>
- [25] A. Einstein, "Quantentheorie des einatomigen idealen Gases," *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, pp. 3–14, 1925.
- [26] Blender Foundation, "Blender - Free and Open 3D Creation Software." *Blender*. Accedit: 28, Nov. 2024. [Online]. Disponible: <https://www.blender.org>
- [27] Adobe, "Adobe - Creative, marketing and document management solutions." *Adobe*. Accedit: 28, Nov. 2024. [Online]. Disponible: <https://www.adobe.com/home>

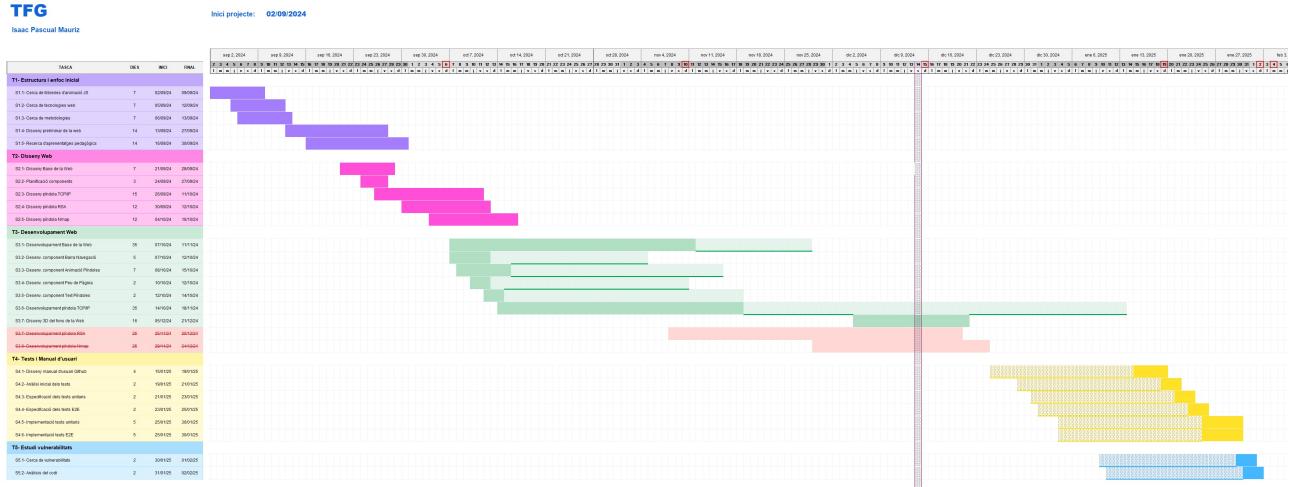


Fig. 3: Diagrama de Gantt.

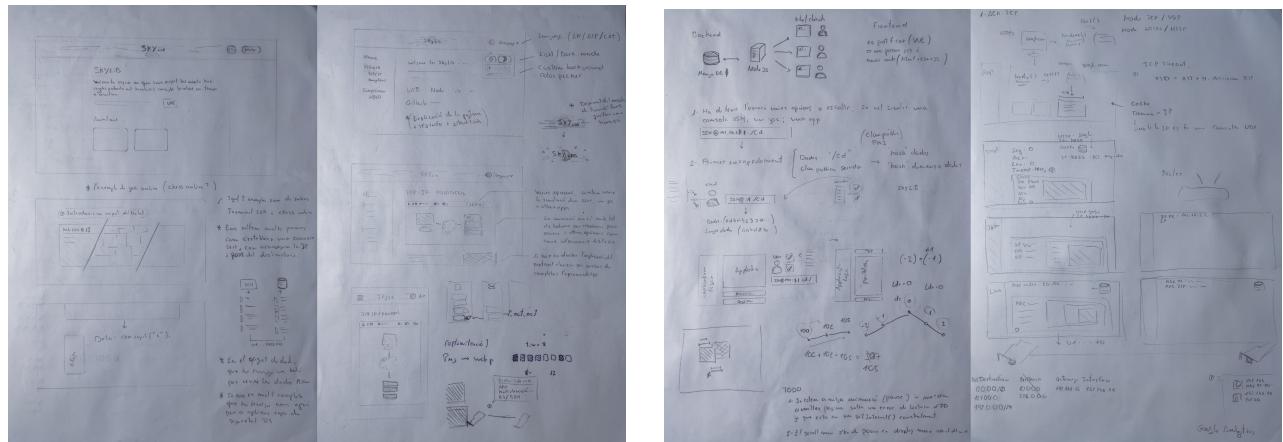
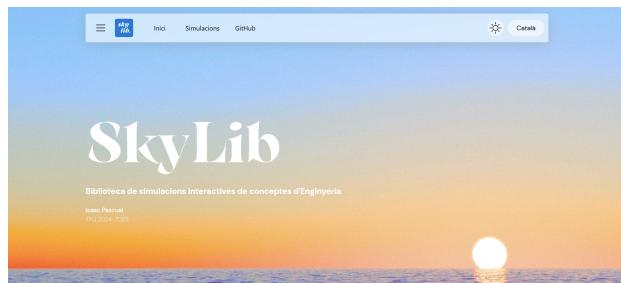
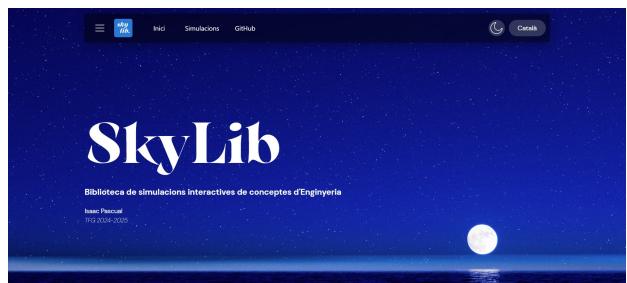


Fig. 4: Recopilació dels prototips wireframe.



(a) Mode diürn (Light Mode)



(b) Mode nocturn (Dark Mode)

Fig. 5: Comparació entre els modes diürn i nocturn de la portada de SkyLib.

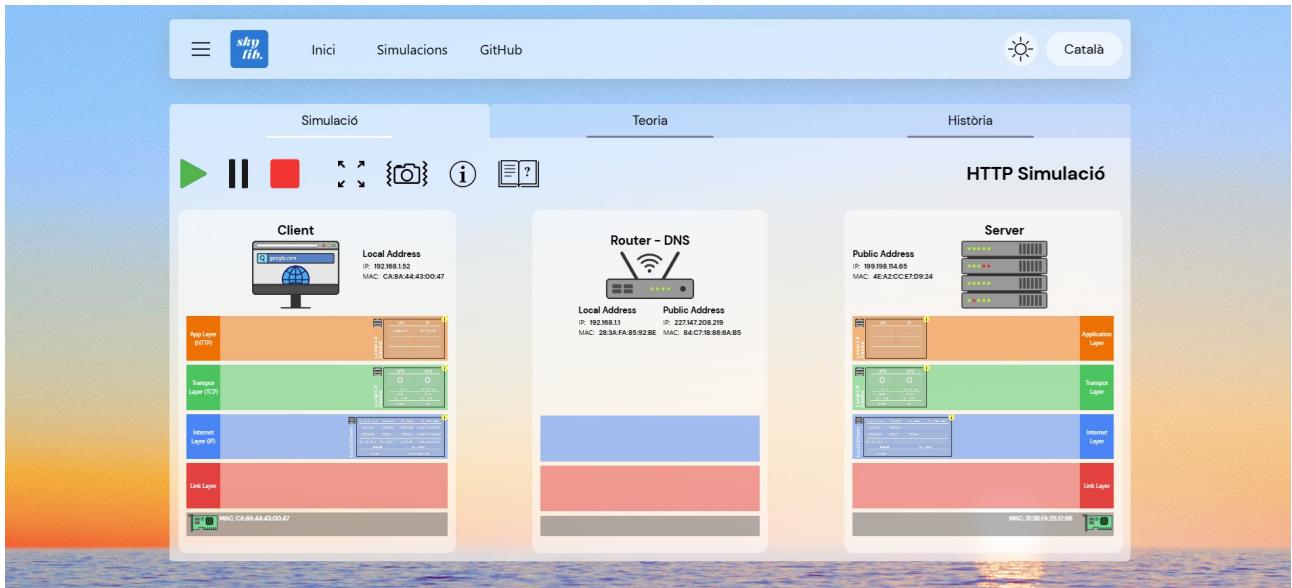


Fig. 6: Captura de la píndola HTTP de la web SkyLib.

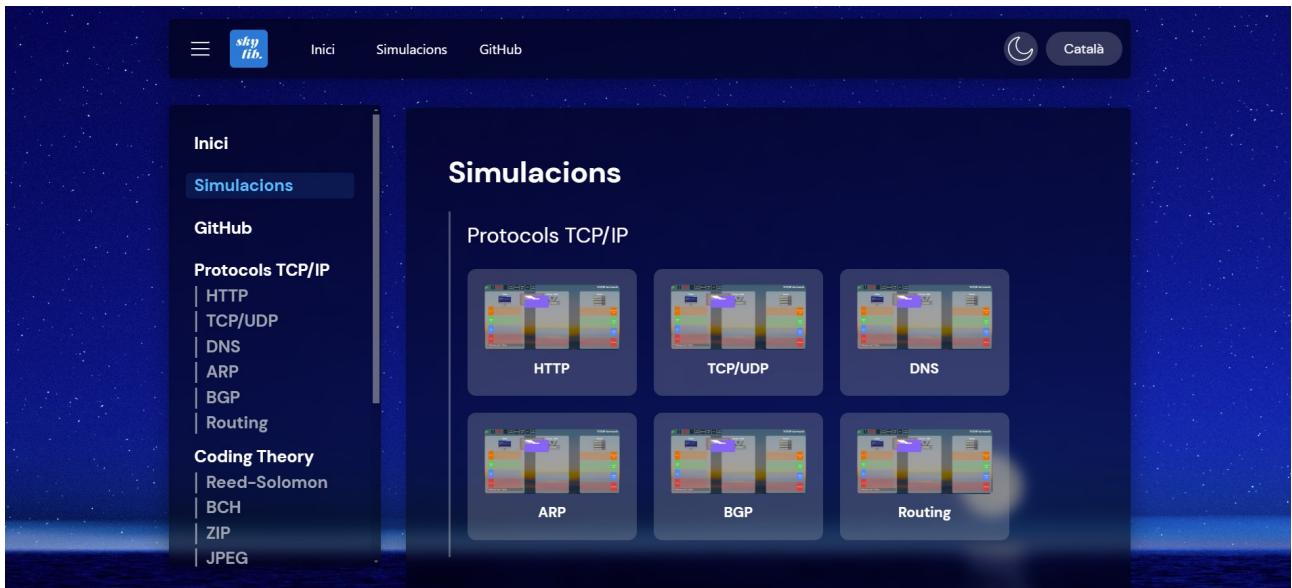


Fig. 7: Captura de l'apartat de simulacions dins la web de SkyLib.

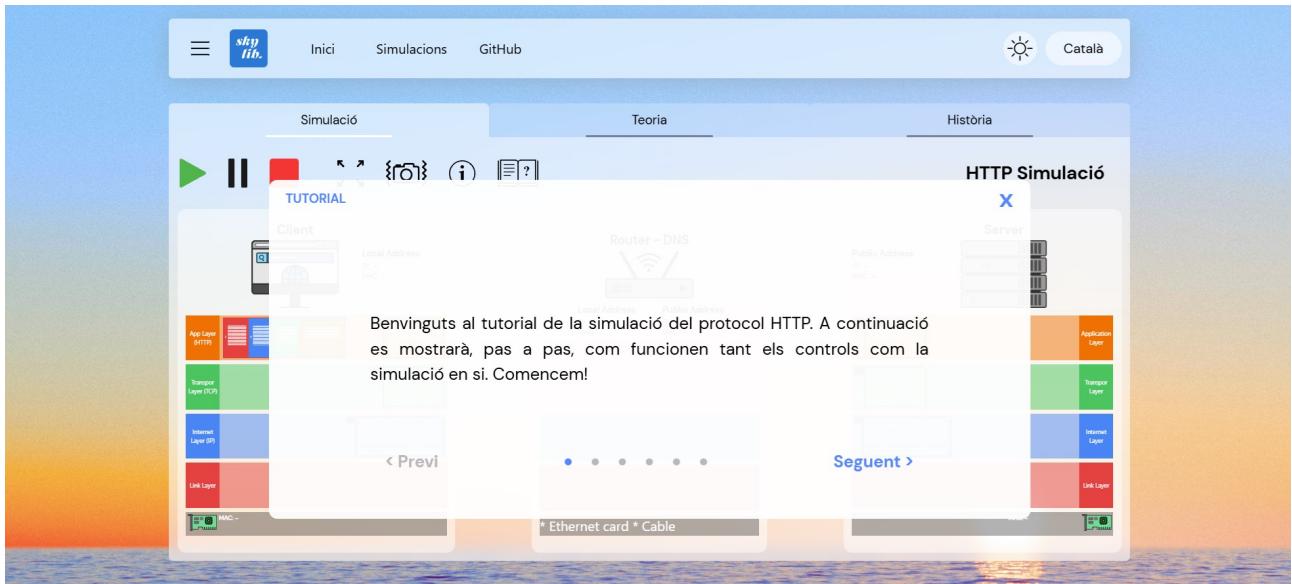


Fig. 8: Captura de l'apartat del tutorial de la píndola HTTP dins la web de SkyLib.

APÈNDIX

A.1 Conceptes bàsics de la metodologia Scrumban i GitHub

En Scrumban tenim una sèrie de terminologies que són necessàries a l'hora de treballar:

- Scrum: És el marc de treball àgil sobre el qual es basa Scrum. Divideix el projecte en iteracions curtes (sprints) per lliurar parts del producte de manera incremental.
- Sprint: Un sprint és una iteració de treball amb una durada fixa (generalment d'1 a 4 setmanes) durant la qual es completen un conjunt de tasques del projecte.
- Product Backlog: Llista completa de funcionalitats, millores o correccions que s'han de fer en el projecte.
- Sprint Backlog: És un subconjunt del product backlog. Aquestes són les tasques que l'equip ha decidit completar durant un sprint.
- Tauler Kanban: És un tauler visual que ajuda l'equip a veure el flux de treball i la situació de cada tasca. Normalment, té columnes que representen els estats de les tasques (per exemple: Per fer, En procés, Fet).
- Cartes Kanban: Cada tasca o ítem de treball és representat per una carta que es mou pel tauler a mesura que avança. Cada carta representa una tasca del backlog que es va movent a través de les diferents fases del treball.

Alguns dels beneficis de la metodologia Scrumban són:

- Flexibilitat: Scrumban permet adaptar-se als canvis en qualsevol moment, es poden afegir noves tasques al backlog tan aviat com s'acaben d'altres, sense haver d'esperar al final d'un sprint.
- Millora contínua del flux de treball: Gràcies al sistema de Kanban, els equips poden veure clarament el progrés de les tasques a través del tauler visual i ajustar el flux de treball segons sigui necessari.
- Control de la càrrega de treball “work in progress”(WIP): Els límits de treball en procés ajuden a evitar que l'equip es sobrecarregui amb massa tasques a la vegada.

En GitHub Flow també existeix una sèrie de terminologies necessàries per treballar:

- Branch (branca): Còpia del codi principal per treballar en tasques separades. Tenim una branca per a cada tasca específica.
- Commit: Canvi registrat al codi amb un missatge descriptiu.
- Push: Enviar els canvis locals a GitHub.
- Pull Request: Sol·licitud per revisar i fusionar canvis d'una branca amb la branca principal.

- Merge: Integrar canvis d'una branca a una altra.
- Fork: Còpia d'un repositori per treballar-hi sense afectar el projecte original.

A.2 Ús de la Suite d'Adobe

Cal destacar l'ús de la suite d'Adobe [27] com a eina principal per al disseny, la maquetació i l'edició dels elements visuals del projecte. Per a la creació de diagrames, il·lustracions i icones, s'ha utilitzat Adobe Illustrator, exportant els gràfics en format SVG, que permet escalar-los sense que es perdi qualitat.

A més, per garantir una coherència visual i evitar possibles problemes de *copyright*, totes les icones i diagrames han estat dissenyats exclusivament per al projecte. En l'àmbit audiovisual, s'ha usat tant Adobe Premiere Pro, Photoshop i Lightroom per al disseny i edició del fons animat, així com el tractament de les imatges i elements gràfics.

Aquesta combinació d'eines ens ha permès aconseguir un estil visual únic i adaptat a l'estètica minimalista del projecte.

A.3 Model d'Objectes del Document (DOM) en HTML

El Document Object Model (*DOM*) és una representació estructurada d'un document HTML o XML en forma d'arbre jeràrquic, on cada element de la pàgina es tracta com un objecte que es pot modificar mitjançant JavaScript i altres tecnologies web.

Quan un navegador carrega una pàgina web, converteix el codi HTML en una estructura DOM, on cada etiqueta HTML es representa com un node dins d'aquest arbre. Això permet als desenvolupadors accedir, manipular i modificar dinàmicament el contingut, l'estructura i l'estil de la pàgina sense necessitat de tornar a carregar-la.

Gràcies a la seva flexibilitat i control el DOM és una de les bases fonamentals per al desenvolupament web modern.

A.4 Selecció de Tecnologies per al Backend

Pel que fa al backend, hi havia múltiples opcions a l'hora d'implementar un servei que proporcionés la pàgina web i gestionés les possibles funcionalitats. Una de les primeres alternatives considerades va ser l'ús d'un servei de hosting dedicat de pagament, el qual permetria alocar un servidor backend amb tecnologies com Node.js, Django o Flask, oferint més flexibilitat i control sobre el sistema.

Una altra opció viable era l'ús de Amazon Web Services (AWS) o altres plataformes de computació al núvol, com Google Cloud Platform (GCP) o Microsoft Azure, que permeten desplegar aplicacions backend de manera escalable i amb alta disponibilitat. Tot i que aquests serveis proporcionen infraestructures robustes, sovint comporten costos recurrents i requereixen una gestió més avançada, factors que no encaixaven del tot amb els objectius d'aquest projecte.

També, aprofitant que compto amb una Raspberry Pi 5, es va valorar la possibilitat de muntar un servidor casolà, una opció econòmica i totalment personalitzable que permetria gestionar el backend de manera autònoma. No obstant això,

aquesta solució implicava la necessitat de mantenir el servidor encès permanentment, gestionar la seguretat i configurar l'exposició del servidor a internet, aspectes que podrien generar més complexitat del necessari per a aquest projecte.

Finalment, vaig optar per desplegar la pàgina web a través de GitHub Pages, una solució gratuïta, pensada per a estudiants i projectes personals. Aquest servei permet allotjar aplicacions estàtiques amb una integració directa amb el control de versions de GitHub, cosa que facilita l'actualització i el manteniment del projecte sense cap cost addicional. D'aquesta manera, GitHub Pages es presentava com l'opció més adequada per a aquest projecte, ja que no requereix gestió de servidors i ofereix una infraestructura fiable per a una aplicació purament frontend.

A.5 Flux d'Execució

Per comprendre el funcionament de l'aplicació, cal analitzar els principals fitxers i components que intervenen en el seu procés d'execució.

La càrrega inicial del projecte comença en el fitxer `index.html`, que actua com a punt d'entrada de l'aplicació. Aquest document inclou l'element HTML `<div id="app">`, que serveix com a contenidor on es muntarà Vue i addicionalment, carregarà el fitxer `main.js` mitjançant un tag `<script>`.

El fitxer `main.js` és el responsable d'inicialitzar l'aplicació. En ell s'importa el component principal `App.vue` i s'estableix la configuració del sistema de rutes mitjançant el fitxer `index.js`, que defineix quines rutes s'han de carregar segons la navegació de l'usuari.

Dins el fitxer `App.vue`, tenim implementada l'estructura base de la interfície web. Aquí es mantenen els elements persistents a totes les pàgines, com la barra de navegació superior, la barra lateral, el footer i els elements del fons. A més, per gestionar la càrrega dinàmica dels components segons la ruta seleccionada, s'utilitza el component de Vue Router `<RouterView>`.

Així, si volem mostrar un component específic, com per exemple el de la píndola HTTP, es pot definir mitjançant un enllaç de navegació `<RouterLink to="/http"></RouterLink>`, que carregarà automàticament el component d'aquell path, sense necessitat de recarregar la pàgina.

A.6 Estructura dels Components

A l'hora de definir un nou component en Vue, s'ha de tenir en compte diversos factors. Un component està format per tres seccions principals:

- `<template>`: Conté l'estructura HTML del component on es defineixen els elements de la interfície d'usuari.
- `<script setup>`: Inclou la lògica en JavaScript que gestiona tant el comportament del component com la seva interacció amb els altres elements.
- `<style>`: Permet afegir estils CSS directament, que s'apliquen únicament al component.

Un altre aspecte rellevant és com es carreguen i criden els components dins de l'aplicació. Primerament, el component s'importa i s'instància dins d'una vista principal, que actua com a contenidor i el gestiona. Quan es renderitza la vista, Vue genera i mostra el component corresponent que es troba encapsulat.

A.7 Exemple de codi per a realitzar una animació

En aquesta secció definirem un exemple bàsic de codi que permet comprendre tant el funcionament del `timeline` com la gestió de les animacions. Primer, es defineix una funció que actuarà de `timeline` dins del `onMounted()`. Aquesta funció s'ha de declarar com a asíncrona `async` per tal de gestionar correctament el flux d'execució amb Promises.

```
async function timeline() {
    input_animation(0, 100, 2000, 1,
    → 'objecte_01_ID');
    await pauseFlow();
    input_animation(0, -100, 1000, 1,
    → 'objecte_01_ID');
    input_animation(50, 50, 3000, 2,
    → 'objecte_02_ID');
    await goto_selected('objecte_02_ID',
    → 1000, 0);
}
```

Primerament, en iniciar el `timeline`, es crida a la funció `input_animation` sobre l'element amb ID `objecte_01_ID`. Aquesta funció aplicarà una transformació en la qual l'element es desplaçarà, durant 2000 mil·lisegons, 0 píxel en l'eix X i 100 píxels en l'eix Y. La seva escala no es modifica, ja que se li passa per paràmetre un valor 1.

En acabar aquesta funció, s'executa automàticament `pauseFlow()`, que pausa l'execució del `timeline`, obligant l'usuari a premer el botó de play.

En prosseguir l'animació, tenim una execució simultània de les dues funcions `input_animation()`, ja que no tenen l'espera explícita `await`. Aquestes apliquen respectivament transformacions en els eixos X i Y en els elements `objecte_01_ID` i `objecte_02_ID`.

Per acabar, ens esperem que l'execució simultània doni pas a la funció `goto_selected()`, la qual ens mourà la càmera cap al `objecte_02_ID` en un transcurs de temps de 1000 mil·lisegons. Addicionalment no se li aplica cap valor lambda, per tant, no variarà la funció logarítmica.