

**uc3m**

Universidad  
**Carlos III**  
de Madrid

Universidad Carlos III  
Sistemas interactivos y Ubicuos  
Course 2024-25

Prototipo de  
kindle controller a.k.a. killer

Date: **22/04/25**

Grupo: **84** Equipo: **13**

Members: **Sonia Navas Rutete, David Sánchez Arranz, Javier Sanz Díaz**

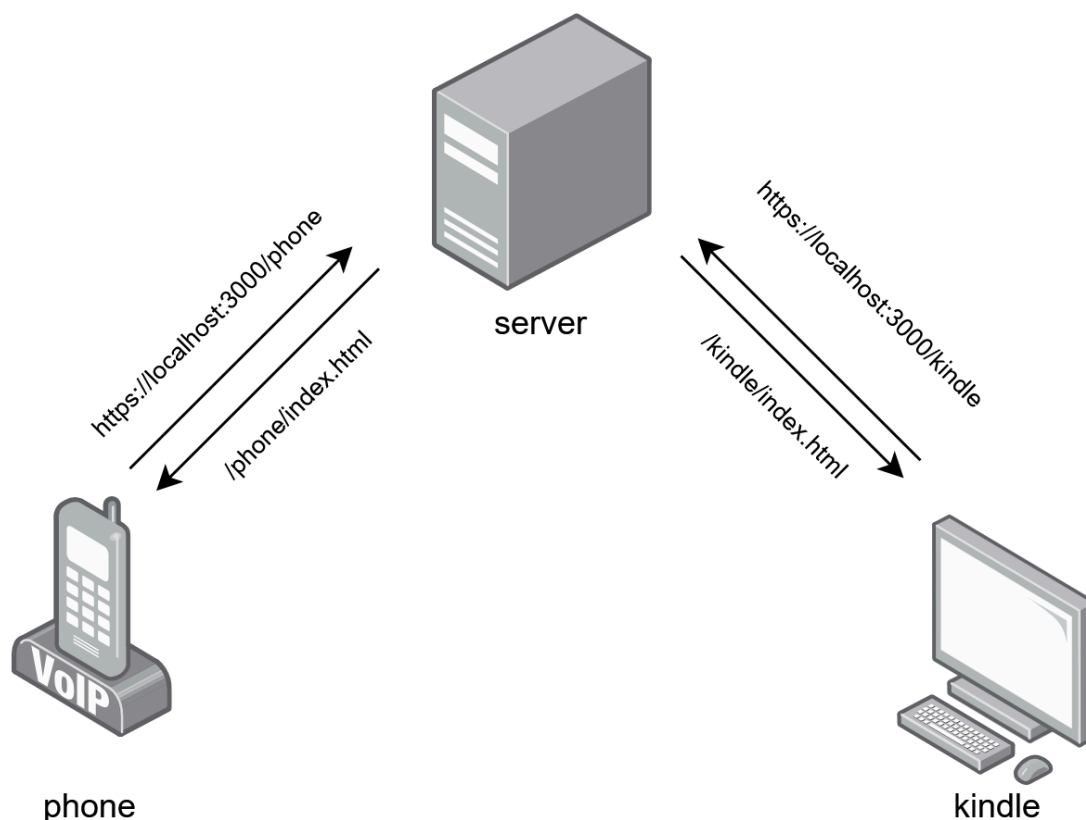
<b>1. Introducción</b>	<b>2</b>
<b>2. Especificaciones técnicas</b>	<b>2</b>
<b>3. Implementación de los prototipos</b>	<b>3</b>
<b>4. Prototipado experiencial y bodystorming</b>	<b>5</b>
<b>5. Iteraciones y mejoras</b>	<b>5</b>
Iteración 1	5
Iteración 2	6
Iteración 3	6
Mejoras	6
<b>6. Reflexión final</b>	<b>7</b>

# 1. Introducción

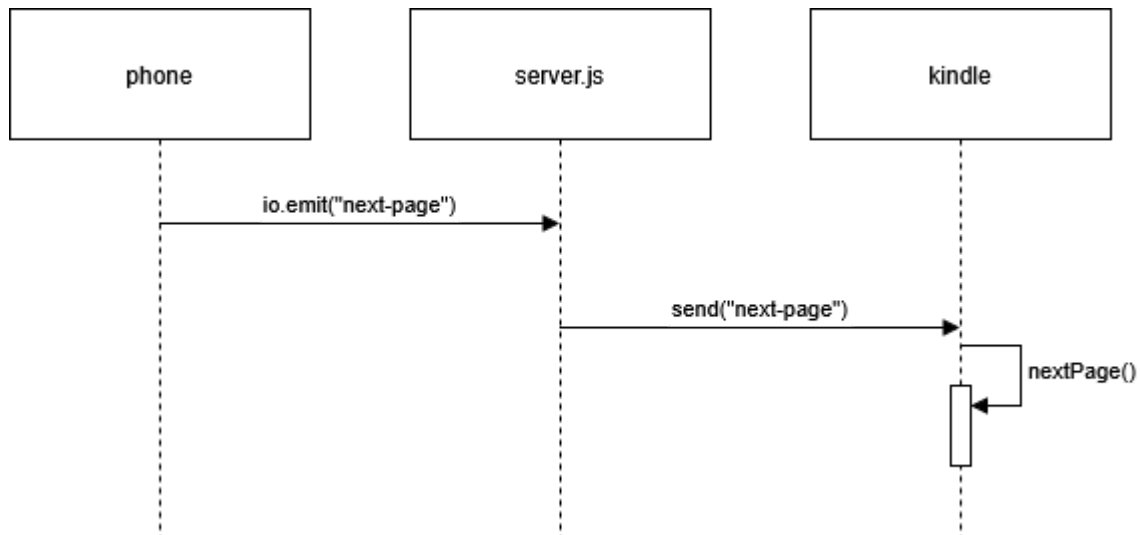
A continuación en el documento se desarrolla sobre los pasos y metodología que hemos usado para desarrollar *Kindle Controller a.k.a. ki..ller*. La propuesta principal es desarrollar un prototipo que a través del móvil pueda controlar un libro electrónico sin necesidad de tener que interactuar directamente con él. Además se han añadido una serie de funcionalidades extra sobre las que hablaremos más adelante para dotar a este prototipo de más utilidad práctica.

## 2. Especificaciones técnicas

Respecto a la arquitectura del sistema hemos hecho uso de Node.js, Express y Socket.IO. Node.js lo hemos empleado para capacitar a nuestro sistema de funcionalidades que puedan trabajar de manera asíncrona. Ha sido un elemento clave para poder gestionar la escucha y emisión de eventos que se han producido por los sockets que gestiona Socket.IO. Express lo hemos usado para crear el servidor y manejar las rutas, aquí podemos ver un diagrama donde explicamos como el servidor https (uso de llave y certificados) en función de la request que efectúes devolverá la interfaz del móvil o del kindle.



A continuación tenemos un esquema que describe una de las comunicaciones básicas que se producen en el sistema, es el evento `onClick(next-page)`.



Podemos ver como desde phone se emite “next-page”, y una vez recibido por el servidor se vuelve a emitir para que lo pueda recibir el kindle que ejecutará la acción que corresponde al paso de página del libro. Los eventos de comunicación de los que consta nuestro sistema, serán expuestos a continuación con sus funcionalidades, son los siguientes:

- next-page: ejecuta el paso de página en el lector de epub del kindle.
- prev-page: ejecuta el paso a la página anterior en el lector de epub del kindle.
- exit-book: si nos encontramos en un libro, este evento de comunicación nos permite volver a la biblioteca.
- turn-off: hace que se active o desactive visualmente el sistema, simulando de esta forma que se ha apagado.
- que-leo: pregunta al kindle cual es el texto que hay que leer.
- texto-leido: trae de vuelta la variable texto que contiene el texto que el móvil va a leer en voz alta.
- pausar-lectura: pausa la lectura en voz alta del móvil.
- pointer-move: simula el movimiento del pointer en el kindle.
- pointer-click: simula un clic en el kindle.

### 3. Implementación de los prototipos

En la etapa de ideación y diseño se propusieron una serie de funcionalidades y modos de implementarlas que a lo largo del proceso de prototipado han ido evolucionando. A continuación, se exponen estas y la justificación de su desarrollo:

- Biblioteca. Se ha implementado una página donde poder ver y seleccionar todos los libros, novelas gráficas o mangas que tengas disponibles para leer. Para volver a la biblioteca desde donde estés se puede realizar un gesto con el móvil hacia arriba o bien interactuar con la interfaz web o móvil.
- Modo Noche. Se ha implementado un estilo alternativo con tonos más oscuros para hacer la lectura más relajada a altas horas de la noche. Para activarlo hay que interactuar con la interfaz web.

- Apagado con botón y automático. Tras un tiempo configurado en el lector, este considera que el lector puede estar inactivo y hace un apagado automático del sistema. También se ha implementado un botón en la interfaz móvil para que se pueda encender y apagar como si fuera un mando de televisión.
- Lector epub. El sistema se ha diseñado para poder leer libros en formato .epub con todos sus detalles (cambio de fuente y tamaño), toda esta implementación la podemos encontrar en la interfaz web.
  - Traductor. Se ha añadido la funcionalidad de traducir fragmentos de texto. Se puede hacer mediante la selección del texto deseado bien a través de la interfaz web o del puntero y seleccionando el botón de traducir. Una vez seleccionado el botón de traducir el texto traducido se guarda en una anotación y se puede acceder a él clickando en el subrayado de color amarillo.
  - Subrayado: Esta funcionalidad se puede activar de forma parecida a la traducción. El color del subrayado es lila.
  - Notas. Las notas funcionan de forma parecida a las dos anteriores. Se selecciona el texto sobre el cual se desea escribir y accionamos el botón de notas. Ahora tenemos que interactuar de manera forzosa con la interfaz web ya que hay que hacer uso del teclado para introducir la nota que queramos, darle al Enter y esta quedará guardada sobre un subrayado de color rosa. La podemos consultar clicando sobre el subrayado.
- Reconocimiento de Voz. Desde el móvil podemos activar y desactivar el reconocimiento de voz y este puede ser usado tanto para navegación como para funcionalidades.
- Para la navegación hemos diseñado varias alternativas.
  - Usar los botones de la interfaz móvil:
    - Botón hacia arriba: saldremos del libro actual
    - Botón derecha/izquierda: navegaremos entre páginas
  - Uso de comandos de voz, para cada comando tenemos diferentes palabras para abarcar un mayor rango de posibilidades:
    - Pasa: pasar página
    - Vuelve: regresa a la página anterior
    - Lee: lee el contenido del capítulo
    - Para de leer: pausa la lectura
  - Movimientos
    - Giro horizontal de 45°: hacia la izquierda pasaría página y hacia la derecha volvería.
    - Movimiento vertical: saldría del libro
- Text to Speech. A través del reconocimiento de voz podemos solicitar que este lea el contenido del capítulo en el que estamos.
- Puntero. Finalmente se ha implementado un touchpad en la interfaz móvil que simula el ratón del ordenador para poder navegar más fácilmente sin necesidad de interactuar con el ordenador, esta permite mover un puntero y hacer click en botones para poder activar el resto de funcionalidades.

A continuación presentamos los vídeos acorde a las funcionalidades explicadas previamente:

- Vídeo mostrando los comandos de voz y la lectura. (<https://youtu.be/4q3ThEw4mqY>)

- Vídeo enseñando los distintos movimientos posibles con el móvil además del apagado del kindle. (<https://youtu.be/C6NjekfSYTk>)
- Vídeo sobre la funcionalidad del puntero y los botones de killer ([https://youtu.be/X5G7NsK\\_ULI](https://youtu.be/X5G7NsK_ULI))
- Vídeo explicativo de las distintas interacciones con el kindle (<https://youtu.be/vFBIPJ4rlS8>)

## 4. Prototipado experiencial y bodystorming

Para elegir cómo controlar los movimientos de las páginas teníamos claro que era necesario tener varias opciones. Para ello simulamos la situación de estar lejos del ebook y cómo le haríamos el gesto a alguien para que pase de página. Surgieron dos opciones y nos quedamos con la que representa el pasar una página en un libro real.

Este gesto funciona bien una vez implementado en el código, pero más importante funciona bien para la persona que lo usa, pues resulta intuitivo.

<https://youtu.be/a8YO1ORZWyo>.

La funcionalidad del audiolibro surge de la nostalgia de que te lean en alto antes de dormir. Cuando estás tan cansado que no quieres ni moverte pero quieres seguir leyendo. Ojalá pudieras simplemente decir algo y que automáticamente el libro se lea solo. Esto de una manera ligeramente cómica viene representado en el siguiente video.

<https://www.youtube.com/watch?v=hRIOgR6s-iM>

## 5. Iteraciones y mejoras

Este proyecto ha ido sufriendo una evolución constante. Entendimos desde el principio que la primera idea rara vez es la definitiva decidimos entrar en un constante basado en la prueba y error buscando la mejor implementación para la interacción del sistema.

### Iteración 1

Como primera iteración desarrollamos un lector de texto plano pero decidimos pasar a uno que pudiera leer formato EPUB. Explorando las posibilidades y las necesidades reales de los lectores digitales, identificamos que el formato EPUB ofrecía una estructura mucho más rica y versátil (estilos, capítulos, etc.). Lo que realmente nos hizo decantarnos por el cambio fue que todas estas posibilidades se ofrecían desde una librería con una implementación sencilla y documentada. Este fue uno de los cambios más significativos respecto a la idea original. Además implementamos la navegación entre páginas con el uso de movimientos en el móvil.

Esto ha sido un reto, conseguir la calibración adecuada puesto que, hemos querido que el gesto fuera lo más natural posible recordando al de pasar una página y para ello hemos tenido que estar testeando con la frecuencia del gesto (tiempo de reinicio) y los grados de inclinación para que la interacción fuera lo más natural posible.

## Iteración 2

En el segundo ciclo de iteración nos propusimos incorporar una biblioteca al prototipo, de esta forma podemos ofrecer más contenidos y de más tipos. Viendo el resultado creemos que ha merecido la pena porque convierte al prototipo en algo mucho más tangible pudiendo añadir los libros que queramos y teniendo variedad.

También decidimos agregar más funcionalidades pero uno de los problemas que nos hemos encontrado es que al ofrecer funcionalidades avanzadas (como análisis de texto o traducción) teníamos la necesidad de utilizar APIs externas. Muchas de las opciones más potentes resultaron ser de pago, lo cual representaba un obstáculo para nuestro ajustado presupuesto. Decidimos entonces adoptar un enfoque práctico e implementar algunas de las que vimos que había opciones de uso gratuito y funcionaban bien como la de traducción, que implementamos al API de traducción de Google.

## Iteración 3

Finalmente decidimos realizar un último ciclo de iteración en el que se añadiría una especie de touchpad a la interfaz móvil permitiendo así otra alternativa más para navegar de forma sencilla por todo el menú del ebook. Durante el desarrollo de este puntero hemos visto que es realmente cómodo usarlo pero no hemos conseguido desarrollar opciones para que este pueda seleccionar texto de los libros debido a falta de tiempo y conocimiento sobre el lector de EPUB. Creemos que sería una gran mejora para el futuro.

## Mejoras

A lo largo del desarrollo del prototipo también se nos han ido ocurriendo muchas mejoras. Teníamos la visión de implementar un sistema innovador que permitiera a los usuarios seleccionar texto mediante comandos de voz y, acto seguido, dictar notas asociadas a esa selección. La realidad ha resultado ser más cruda, ya que hemos descubierto que hacer esto realmente preciso resulta un desafío considerable y que la herramienta de reconocimiento de voz no es tan precisa como se esperaba. Aun así creemos que con más tiempo podríamos desarrollar más la interacción por voz ya que resulta muy interesante sobre todo para este prototipo.

Otras idea que nos ha surgido es crear un chatbot Interactivo que pueda interactuar con el usuario sobre el libro que está leyendo. Esto podría incluir desde responder preguntas sobre el contenido hasta generar quizzes o juegos basados en la trama o personajes.

También nos hubiera gustado darle una dimensión social a nuestro prototipo, permitiendo a los usuarios conectar con amigos, compartir su progreso de lectura, intercambiar recomendaciones o incluso comentar pasajes específicos de los libros de forma conjunta.

Finalmente también creemos que se podría desarrollar una gestión más evolucionada de las anotaciones. No solo incluir notas y subrayado, sino que ofrecer la posibilidad de buscar, organizar tus notas, exportarlas o compartirlas con otras personas.

## 6. Reflexión final

Este trabajo nos ha hecho pasar por las diferentes fases del duelo: negación, ira, negociación, depresión y aceptación. Antes de llegar al duelo estábamos emocionados. Tantas ideas, tantas opciones, infinitas casi. Cada uno pensaba en lo que quería hacer y todas las funcionalidades que quería implementar. Hasta llevábamos varias horas intentando implementar una “tontería” y no funcionaba. Ahí empezaba la negación, que venía de la mano con la ira “Estoy haciendo mal, esto no debería ser tan difícil, es una tontería, a ver que estás haciendo, déjame que en una hora lo tengo.”

En ocasiones no pasaba de ahí, un poco de ira y acaba funcionando. Pero otras veces, demasiadas, hemos tenido que optar por la negociación. “No funciona perfecto pero cumple casi todos los casos”, “es un poco feo pero al fin y al cabo funciona”, “en cuanto consiga que se entienda la intención lo dejo”. La depresión va intrínseca en el estudiante universitario así que directamente llegamos a la aceptación. Se han llevado a cabo más funcionalidades de las que en un principio pensábamos. No funciona todo pero es normal. Porque es un prototipo.

Ese ha sido el fallo que nos ha perseguido a lo largo de las semanas. Es un prototipo y no hace falta que tenga un nivel de detalle para ser comercializado o para ser minuciosamente juzgado. Nos ha costado darnos cuenta de qué límites tendríamos que poner. Hasta qué punto en el desarrollo tenemos que llegar. Pero al final hemos sido capaces de hacerlo y aquí tenemos nuestro prototipo.

Con respecto al ámbito técnico y la organización del proyecto, además de usar git como herramienta colaborativa, hemos aprendido a usar sus ramas, mejorando la distribución y gestión del trabajo. Al igual que las distintas APIs para realizar las diferentes funciones implementadas, descubriendo la importancia del ensayo y error para calibrar estas funcionalidades a su ejecución óptima.

Más allá de esto, sentimos que hemos construido una buena base con este prototipo para avanzar a la siguiente fase.