



chuOK

Alumno: Samuel Tineo Herrera

Curso: 2024/2025

ÍNDICE

1. Introducción y presentación del proyecto
2. Diseño físico y lógico de la red
3. Requisitos hardware y software
4. Análisis y diseño de la aplicación
 - a. Análisis de requisitos
 - b. Diseño del modelo de datos:
 - i. Modelo Entidad/Relación
 - ii. Modelo relacional
 - iii. Análisis de las formas normales, mínimo 3FN
 - iv. Representación de la base de datos en el SGBD elegido
 - c. Diseño del sitio web:
 - i. Mapa del sitio web, indicando las páginas existentes y los enlaces entre estas
5. Proceso de creación
6. Seeding de contenido y lógica de funcionamiento del modo aventura
7. Modo libre
8. Elementos públicos
9. Manual del usuario
10. Manual técnico (destacando los puntos más interesantes de la aplicación)
11. Propuesta de ampliación del proyecto
12. Estudio de viabilidad del proyecto
13. Conclusiones
14. Bibliografía
15. Anexos

1. Introducción y presentación del proyecto

En el vertiginoso mundo actual, donde el ritmo de vida parece no conceder tregua y el estrés laboral se ha normalizado hasta convertirse en un compañero constante, nace chuOK, una aplicación web concebida como un refugio digital, una pausa consciente en medio del caos cotidiano. El nombre chuOK proviene del término coreano "추억" (pronunciado "chu-eok"), que hace referencia a los recuerdos o memorias, generalmente agradables y nostálgicas. Esta elección no es casual: evoca la idea de reconectar con uno mismo, de volver a esos momentos de calma y satisfacción personal que solemos relegar por falta de tiempo o energía. Además, se han destacado visualmente las letras "O" y "K" dentro del nombre para enfatizar el reconocido símbolo de aprobación del inglés, "OK", sugiriendo bienestar, conformidad y serenidad, que son precisamente los valores que la plataforma busca transmitir.

chuOK es mucho más que una simple herramienta de entretenimiento. Es una propuesta consciente para ofrecer al usuario una vía de escape saludable, donde pueda desconectar de la presión constante del trabajo, los plazos, las responsabilidades domésticas o académicas, y reconectar con el placer de aprender y descubrir. En un panorama digital saturado de contenido pasivo, en el que predomina el consumo automático de vídeos cortos, series interminables y plataformas de streaming diseñadas para mantener la atención secuestrada el mayor tiempo posible, chuOK se desmarca con una filosofía clara: ofrecer entretenimiento activo. Es decir, contenido que no solo entretenga, sino que también estimule intelectualmente al usuario, que lo rete de forma amena, que despierte su curiosidad y le aporte conocimientos nuevos mientras se divierte.

La mayoría de las aplicaciones actuales que prometen "desconectar" terminan promoviendo una forma de evasión que, si bien puede resultar relajante a corto plazo, rara vez deja una huella positiva o duradera. Frente a esta tendencia, chuOK propone un enfoque completamente distinto. En lugar de fomentar el consumo pasivo, incentiva al usuario a participar activamente en su propio proceso de descanso, convirtiendo ese tiempo libre en una experiencia enriquecedora. La plataforma está cuidadosamente diseñada para ofrecer una selección de frases célebres, artículos breves e inspiradores, curiosidades históricas, retos mentales y pequeños rompecabezas, todos ellos pensados para estimular la mente de forma ligera pero significativa.

Esta propuesta no es fruto del azar. Tras un periodo de investigación personal y observación del entorno profesional, he identificado una problemática que afecta especialmente al sector del desarrollo de software: el alto índice de agotamiento (burnout) entre los profesionales. Esta disciplina, que suele asociarse con salarios elevados y ciertas comodidades laborales, esconde una realidad mucho más compleja. Detrás del código y las interfaces hay jornadas extensas, presión constante por cumplir con entregas, cambios de última hora en los requerimientos, reuniones interminables y una exigencia permanente de actualización técnica. Todo ello, sumado a la naturaleza sedentaria del trabajo, genera un caldo de cultivo perfecto para el deterioro de la salud mental.

El burnout o síndrome de agotamiento profesional es un trastorno psicológico relacionado con el entorno laboral que se caracteriza por un estado de estrés crónico, agotamiento emocional y despersonalización. Se reconoció oficialmente como un fenómeno laboral por la Organización Mundial de la Salud (OMS) en 2019 y está clasificado en la CIE-11 (Clasificación Internacional de Enfermedades) como un fenómeno ocupacional, no como una condición médica.

Más información en el Anexo 1

chuOK nace también como una respuesta a esa necesidad específica dentro del sector tecnológico. Aunque su alcance es global y está abierto a cualquier tipo de usuario, la aplicación pone un foco especial en los desarrolladores de software, un colectivo al que pertenezco y al que me dirijo con especial empatía. Estoy convencido de que las personas que se dedican a esta área comparten una característica común: la curiosidad. Son individuos a los que les gusta entender cómo funcionan las cosas, resolver problemas, enfrentarse a nuevos desafíos. Por eso, el formato interactivo de la plataforma no solo atraerá su atención, sino que además les motivará a seguir explorando y aprendiendo, convirtiendo el tiempo de descanso en una experiencia positiva, activa y nutritiva.

El propósito central de chuOK es claro: entretener. Pero el medio para alcanzar este objetivo es lo que realmente marca la diferencia. Mientras las redes sociales y las plataformas de vídeo apuestan por la recompensa instantánea y la permanencia sin esfuerzo, chuOK plantea el entretenimiento como una travesía intelectual, una pequeña aventura personal en la que el usuario tiene un rol activo. Desde el momento en que accede a la aplicación, se le invita a participar en un proceso de descubrimiento progresivo. De hecho, el contenido inicial está limitado: los nuevos usuarios solo tienen acceso al primer capítulo, lo que les impulsa a avanzar para desbloquear el resto. Así, se establece una dinámica de exploración y recompensa que mantiene el interés y la motivación sin recurrir a mecanismos adictivos.

La plataforma está estructurada en dos grandes secciones. La primera es una zona más relajada, ideal para momentos breves de desconexión, donde el usuario encontrará una amplia variedad de frases célebres, citas inspiradoras, artículos breves y datos curiosos que puede leer de forma aleatoria o seleccionada. La segunda sección, denominada modo aventura, es donde reside el núcleo interactivo de la experiencia. Aquí, el usuario se adentra en un universo gamificado, dividido en mundos temáticos. Para avanzar de un mundo al siguiente, deberá superar diferentes retos: preguntas de opción múltiple, acertijos, pruebas de lógica y otros ejercicios que requieren atención y razonamiento. Cada mundo consta de 20 niveles, y solo completándolos podrá desbloquear el siguiente, generando así una sensación de progreso y logro.

Además, todos los avances y estadísticas del usuario estarán disponibles para su consulta en el home, con un resumen visual accesible de sus logros y progreso general. Para quienes deseen una visión más detallada, la sección Account dentro del User Center ofrece un desglose completo de métricas, tiempo de uso, contenidos completados, niveles desbloqueados, y mucho más. Esta información permite al usuario llevar un seguimiento

consciente de su evolución dentro de la plataforma, y celebrar sus pequeños logros personales, que aunque no sean profesionales, tienen un impacto directo en su bienestar emocional y mental.

En resumen, chuOK es una aplicación pensada para ofrecer un espacio de entretenimiento alternativo, donde el descanso se convierte en una oportunidad para aprender, pensar, explorar y reencontrarse con el placer de la curiosidad. Una plataforma que pone en valor el tiempo del usuario y lo transforma en una experiencia activa, enriquecedora y, sobre todo, humana.

2. Diseño físico y lógico de la red

3. Requisitos hardware y software

A continuación, se detallan los requisitos de hardware y software necesarios para los diferentes entornos implicados: desarrollo, ejecución (cliente) y un posible entorno futuro de despliegue/producción.

a. Requisitos software

Entorno de desarrollo:

El entorno de desarrollo se basa en un sistema operativo Windows, utilizando herramientas modernas de desarrollo frontend y backend. Las tecnologías y herramientas empleadas son las siguientes:

- Sistema operativo: Windows 11 Home (64 bits)
- Editor de código: Visual Studio Code 1.89 o superior
- Frontend: Angular 17, construido con Node.js y Angular CLI
- Backend: Spring Boot 3 (Java 17)
- Base de datos: H2 para desarrollo local (base de datos embebida)
- Control de versiones: Git (integrado con GitHub)
- Gestor de dependencias frontend: npm 10.x
- Entorno de ejecución backend: JDK 17, Maven

- Terminal de desarrollo: PowerShell

Requisitos cliente (usuario final)

Para acceder a la aplicación, el usuario solo necesita un navegador moderno. La aplicación está diseñada como SPA (Single Page Application), por lo que no se requieren instalaciones adicionales:

- Navegadores compatibles:
 - Google Chrome (últimas 2 versiones)
 - Firefox
 - Microsoft Edge
 - Safari (iOS/macOS)
- Sistema operativo cliente:
 - Windows 10/11
 - macOS
 - Android 9 o superior
 - iOS 14 o superior

Requisitos de red: conexión estable a internet; no se requiere ancho de banda elevado.

b. Requisitos hardware:

Equipo de desarrollo

El desarrollo se realiza sobre un equipo doméstico con las siguientes características técnicas mínimas recomendadas para garantizar un flujo de trabajo fluido, especialmente dado que Angular y Spring Boot son entornos exigentes en recursos:

- Procesador: Intel Core i5 de 10ª generación o superior / AMD Ryzen 5
- Memoria RAM: 8 GB mínimo, 16 GB recomendados
- Almacenamiento: mínimo 50 GB libres en disco SSD
- Resolución de pantalla: 1920x1080 (Full HD) para trabajar cómodamente con Angular, navegador y consola abiertos simultáneamente

Entorno de ejecución (servidor en producción)

– *NO DEFINIDO AÚN*

4. Análisis y diseño de la aplicación

a. Análisis de requisitos

La aplicación chuOK tiene como objetivo proporcionar un espacio digital en el que los usuarios puedan desconectar del estrés diario mediante una experiencia de entretenimiento activo y enriquecedor. Con esto en mente, se identifican los siguientes requisitos funcionales y no funcionales, los cuales pasaremos a detallar a continuación:

Requisitos funcionales (RF): Los requisitos funcionales describen las funciones concretas que debe realizar el sistema. Es decir, detallan las acciones que podrá llevar a cabo el usuario y el comportamiento que se espera de la aplicación en respuesta a dichas acciones. Algunos ejemplos comunes son el registro de usuarios, la navegación entre secciones, o la interacción con contenidos. En chuOK podemos encontrar la siguiente selección:

- RF1: El usuario podrá registrarse y acceder con su cuenta. A través de dos formularios; signup, donde podrá registrarse escribiendo un nombre de usuario válido, además de un correo electrónico y una contraseña. Por otro lado se encuentra el formulario de login, donde el usuario deberá facilitar el correo electrónico y contraseña usados en el registro para acceder a la aplicación.
- RF2: El usuario podrá acceder a dos modos principales: Librería, que podríamos llamar “Modo relajado” y Modo aventura. Esto lo hará a través de la vista Home, a la que el usuario se le ofrece al loguear su cuenta en la aplicación. Aquí tendrá un recuadro con dos opciones y este podrá clicar en cualquiera de ellas.
- RF3: En el Modo relajado, el usuario podrá leer frases célebres, artículos breves y curiosidades. Estructurados como una cartelera en forma de rejilla, el usuario podrá ir viendo un pequeño recuadro con un adelanto de cada artículo o frase, y clicando sobre cualquiera de estos, se abrirá una ventana en la que se encontrará el contenido deseado.
- RF4: En el Modo aventura, el usuario deberá completar desafíos para avanzar entre niveles y mundos. La lógica es la siguiente: El usuario comienza en un mundo y los otros dos están bloqueados inicialmente, de esta manera, el usuario deberá superar todos los obstáculos del primer mundo para pasar al siguiente.
- RF5: El sistema guardará el progreso del usuario en cada modo. Se prevé una tabla en la base de datos llamada “Progreso”, que guardará si un usuario ha completado un nivel, y si es así cuándo. Actualizando así, una estadística de progreso de cada mundo, que se ofrecerá directamente en la vista correspondiente.
- RF6: El usuario podrá consultar sus estadísticas generales en su perfil. Para mayor extensión y detalle de estadísticas, se ofrece un apartado en el User Center, dentro de

Account, donde el usuario podrá observar algunas de estas más específicamente (número de niveles completados, tiempo de juego, cantidad de intentos empleados, etc.).

- RF7: Se debe restringir el acceso a ciertos contenidos si no se ha completado el nivel anterior. Como se mencionó anteriormente, una parte crucial de lo que hace al modo aventura diferente es la motivación de poder desbloquear nuevos mundos con una temática diferente, dando al usuario la sensación de que realmente su esfuerzo por completar niveles está dando frutos, y no ha sido en vano.

Requisitos no funcionales (RNF): Los requisitos no funcionales hacen referencia a las características de calidad del sistema, como el rendimiento, la seguridad, la escalabilidad, la accesibilidad o la compatibilidad. No describen lo que el sistema hace, sino cómo lo hace o bajo qué condiciones debe operar. Podemos destacar los siguientes:

- RNF1: La aplicación debe ser accesible desde navegadores modernos. Al ser una aplicación web, el navegador es nuestra llave de acceso, por lo que es de suma importancia mantener el navegador actualizado, no solo para mantenernos seguros y al día, sino también para evitar fallos de compatibilidad con aplicaciones web como chuOK.
- RNF2: El tiempo de respuesta del sistema debe ser inferior a 2 segundos en cualquier operación habitual. Dada la tecnología actual, el usuario no debería quedarse esperando por cada petición que haga. La elección de Angular para construir el front-end de la aplicación viene a solucionar esta problemática, pues su modelo SPA (Single Page Application) nos permite renderizar las vistas de manera instantánea, reduciendo así el tiempo de espera al mínimo.
- RNF3: La interfaz debe ser clara, visualmente atractiva y responsive. Todos sabemos de la influencia de los colores en nuestra percepción de las cosas, chuOK mantiene una paleta de colores a lo largo de toda la aplicación que favorece la tranquilidad y la concentración, factores clave para pasar un rato entretenido y, a su vez, productivo. Cabe destacar la responsividad total de todas las vistas que componen la aplicación, para que visitarla sea una experiencia inolvidable, sin importar el navegador, el dispositivo o la resolución de pantalla desde la que se accede.
- RNF4: El sistema debe ser seguro en cuanto a autenticación y almacenamiento de datos. Quizá el tema más importante para ganarse la confianza del usuario es que este sepa que sus datos están seguros a ciencia cierta. Para ello chuOK pone a su disposición todo lo que ofrece Spring Security para garantizar la encriptación de datos, autenticación y permisos de acceso, etc.

b. Diseño del modelo de datos:

Antes de diseñar la estructura de la base de datos que dará soporte a la aplicación chuOK, es importante comprender algunos conceptos fundamentales relacionados con el modelado de datos. Estos conceptos permiten representar la información de manera organizada, coherente y optimizada para su almacenamiento y posterior consulta.

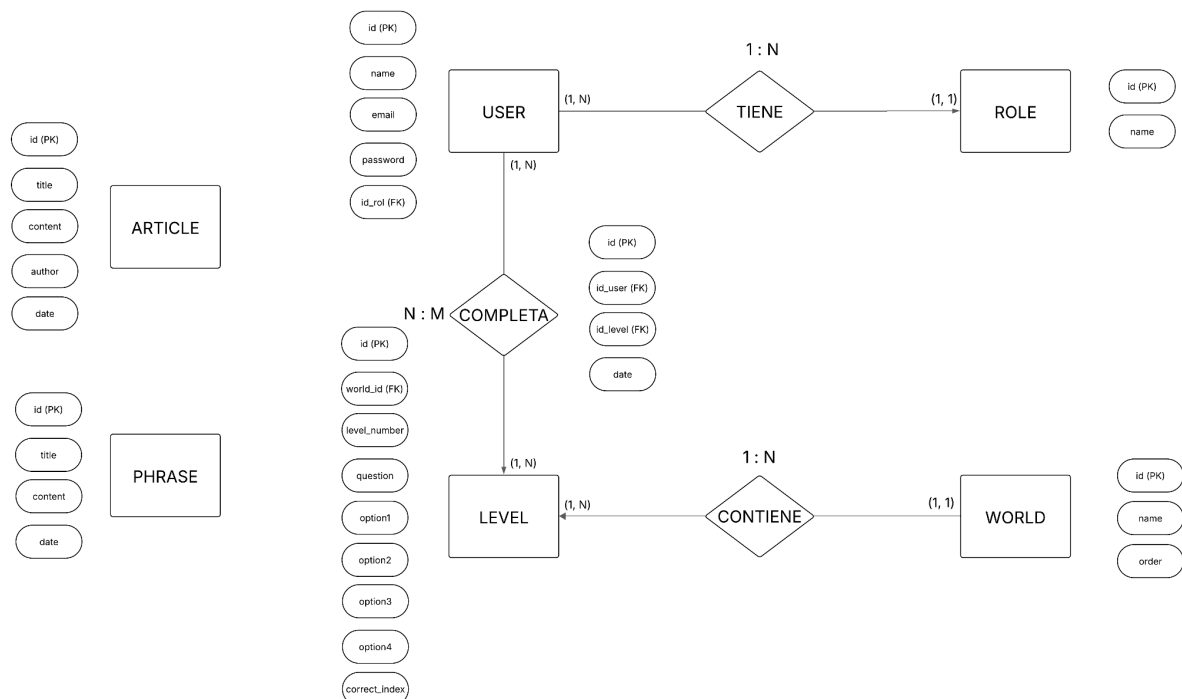
i. Modelo Entidad/Relación

El modelo Entidad-Relación es una herramienta conceptual utilizada en la fase de análisis para representar gráficamente los elementos principales de una base de datos y las relaciones entre ellos.

En este modelo, se representan:

- Entidades: objetos o conceptos relevantes del sistema (por ejemplo, Usuario, Nivel, Artículo).
- Atributos: propiedades de cada entidad (por ejemplo, el nombre o el email del usuario).
- Relaciones: asociaciones entre entidades (por ejemplo, un usuario completa niveles).

Este modelo se representa habitualmente mediante un diagrama E/R, que facilita la comprensión global de la estructura de los datos antes de su implementación. A continuación se ofrece un diagrama E/R basado en las entidades propias de la aplicación:



PK: Primary Key
FK: Foreign Key

I : N: Relación de uno a muchos
N : M: Relación de muchos a muchos

En este diagrama vemos:

- USER

Esta tabla representa a los usuarios del sistema.

- id (PK): Clave primaria única para identificar al usuario.
- name: Nombre del usuario.
- email: Correo electrónico del usuario.
- password: Contraseña cifrada del usuario.
- id_rol (FK): Clave foránea que hace referencia a la tabla ROLE, indicando el rol asignado al usuario.

- ROLE

Define los distintos roles que puede tener un usuario (por ejemplo, administrador, estudiante, profesor).

- id (PK): Clave primaria del rol.
- name: Nombre del rol (e.g., "admin", "student").

- WORLD

Representa un mundo o escenario dentro del sistema, como parte de una estructura gamificada.

- id (PK): Identificador único del mundo.
- name: Nombre del mundo.
- order: Orden o secuencia del mundo (útil para establecer progresión).

- LEVEL

Cada nivel pertenece a un mundo y contiene una pregunta con opciones de respuesta.

- id (PK): Identificador del nivel.
- world_id (FK): Clave foránea que apunta al mundo al que pertenece el nivel.
- level_number: Número que indica la posición del nivel dentro del mundo.
- question: Pregunta del nivel.

- option1, option2, option3, option4: Opciones de respuesta.
- correct_index: Índice de la opción correcta (por ejemplo, 1, 2, 3 o 4).

- **COMPLETA**

Es una tabla intermedia para modelar la relación N:M entre USER y LEVEL. Registra cuándo un usuario ha completado un nivel.

- id (PK): Clave primaria.
- id_user (FK): Clave foránea al usuario que completó el nivel.
- id_level (FK): Clave foránea al nivel completado.
- date: Fecha en que se completó el nivel.

- **ARTICLE**

Contiene artículos que podrían servir como contenido educativo o informativo.

- id (PK): Identificador del artículo.
- title: Título del artículo.
- content: Contenido del artículo.
- author: Autor del artículo.
- date: Fecha de publicación.

- **PHRASE**

Probablemente usada para mostrar frases motivacionales o educativas.

- id (PK): Identificador de la frase.
- title: Título o encabezado de la frase.
- content: Contenido del texto de la frase.
- date: Fecha de inclusión en el sistema.

Relaciones entre tablas:

- **TIENE (1:N entre ROLE y USER)**
Un ROLE puede ser asignado a muchos USERS.

Cada USER tiene un solo ROLE.

Representa una relación 1:N (uno a muchos) desde ROLE hacia USER.

- CONTIENE (1:N entre WORLD y LEVEL)
Un WORLD contiene muchos LEVELS.

Cada LEVEL pertenece a un único WORLD.

Relación jerárquica que organiza los niveles dentro de mundos.

- COMPLETA (N:M entre USER y LEVEL)
Un USER puede completar muchos LEVELS.

Un LEVEL puede ser completado por muchos USERS.

Relación N:M con una entidad intermedia (COMPLETA) que también almacena la fecha de completación.

De esta manera, podemos hacer un resumen del flujo de información de la base de datos:

- Usuarios (USER) tienen un rol (ROLE).
- Los usuarios completan niveles (LEVEL).
- Los niveles pertenecen a un mundo (WORLD).

Además del sistema de niveles y mundos, hay contenido adicional como:

- Artículos (ARTICLE): para lectura.
- Frases (PHRASE): para motivación o enseñanza.

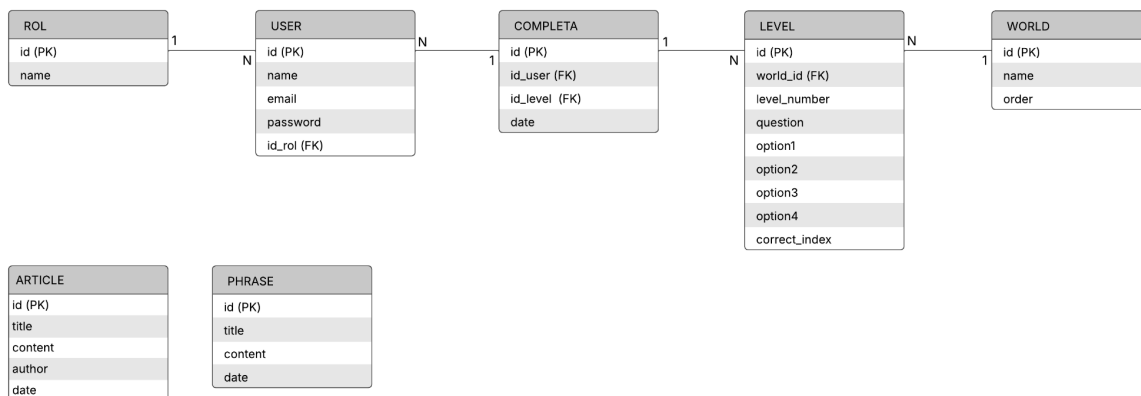
ii. Modelo relacional

El modelo relacional es una transformación del modelo E/R a una estructura de tablas, que puede ser implementada directamente en un sistema gestor de bases de datos relacional (SGBD).

Cada entidad del modelo E/R se convierte en una tabla, y sus atributos se transforman en columnas. Las relaciones entre entidades se representan mediante claves foráneas (foreign keys) que enlazan registros entre tablas.

Este modelo proporciona una visión lógica de la base de datos y sirve como base para su implementación física.

De la misma manera, debajo se ofrece un modelo relacional basado en la estructura de las tablas definidas en el diagrama entidad-relación que componen la base de datos de chuOK:



Tanto para el diagrama Entidad-Relación como para el diagrama relacional se ha utilizado la herramienta Lucidchart.

Lucidchart es una herramienta profesional de diagramación en línea, usada ampliamente en empresas, universidades y equipos de desarrollo. Ofrece una experiencia fluida y colaborativa, especialmente para equipos.

Más información en el Anexo 3

iii. Análisis de las formas normales, mínimo 3FN

El proceso de normalización consiste en aplicar una serie de reglas o "formas normales" para organizar los datos de manera eficiente, evitar redundancias y asegurar la integridad de la base de datos. Las tres primeras formas normales son las más utilizadas en la práctica:

- **Primera Forma Normal (1FN):**
Se cumple cuando todos los atributos contienen valores atómicos, es decir, no hay listas o conjuntos en una sola celda.
- **Segunda Forma Normal (2FN):**
Se cumple cuando ya se tiene la 1FN y no existen dependencias parciales en tablas con claves primarias compuestas. Cada atributo debe depender de toda la clave primaria, no de una parte de ella.
- **Tercera Forma Normal (3FN):**

Se cumple cuando ya se tiene la 2FN y no hay dependencias transitivas, es decir, ningún atributo debe depender de otro atributo que no sea clave.

Aplicar estas formas normales mejora la calidad del diseño de la base de datos, facilita las consultas, y reduce errores por inconsistencias en los datos. A continuación se detalla cada tabla con sus componentes, para un estudio exhaustivo de la relación que existe entre los distintos elementos de cada una.

A continuación, se comprueba que todas las tablas cumplen al menos hasta la Tercera Forma Normal (3FN):

- Primera Forma Normal (1FN)

Se cumple porque:

- Todos los atributos contienen valores atómicos, es decir, no hay listas ni múltiples valores por celda.
- No se repiten grupos de columnas relacionados.

- Segunda Forma Normal (2FN)

Se cumple porque:

- Ninguna tabla tiene claves primarias compuestas (todas usan un solo campo como clave primaria).

Por lo tanto, no existen dependencias parciales (un atributo que dependa de parte de una clave).

- Tercera Forma Normal (3FN)

Se cumple porque:

- Todos los atributos no clave dependen directamente de la clave primaria.
- No hay dependencias transitivas. Por ejemplo:
En la tabla Usuario, el rol_id es una clave foránea, pero no contiene atributos del rol (como el nombre del rol), por lo que no hay dependencia indirecta.

En Nivel, todo depende solo del id del nivel, y el mundo al que pertenece está correctamente separado.

iv. Representación de la base de datos en el SGBD elegido

La base de datos utilizada para la aplicación ha sido implementada con MySQL, un sistema de gestión de bases de datos relacional ampliamente adoptado en entornos profesionales y académicos. Para facilitar la configuración, portabilidad y aislamiento del entorno, se ha optado por ejecutar MySQL dentro de un contenedor Docker.

Este enfoque permite simular un entorno similar al de producción, facilita el mantenimiento y reduce posibles errores relacionados con discrepancias entre máquinas de desarrollo.

Docker es una plataforma de código abierto que automatiza la creación, despliegue y ejecución de aplicaciones mediante el uso de contenedores.

Un contenedor es una unidad ligera y portátil que encapsula todo lo necesario para ejecutar una aplicación: código, runtime, bibliotecas, dependencias y configuración.

Más información en el Anexo 4

El contenedor de MySQL ha sido creado utilizando una imagen oficial desde Docker Hub. A continuación, se muestra un ejemplo del comando de creación:

```
PS C:\Users\stine> docker run --name mysql-chuok-db -e MYSQL_ROOT_PASSWORD=contraseña -e MYSQL_DATABASE=chuok -e MYSQL_USER=chuok_user -e MYSQL_PASSWORD=contraseña_usuario -p 3306:3306 -d mysql:8.0
```

- `MYSQL_ROOT_PASSWORD`: Contraseña del usuario root.
- `MYSQL_DATABASE`: Base de datos inicial creada automáticamente.
- `MYSQL_USER` y `MYSQL_PASSWORD`: Usuario específico para la aplicación.
- `-p 3306:3306`: Expone el puerto para poder conectarse desde la máquina host o desde aplicaciones externas como Spring Boot o clientes MySQL.
- `-d mysql:8.0`: Versión de la imagen MySQL a utilizar.

La base de datos se gestiona y visualiza desde la terminal utilizando el cliente oficial de MySQL, o mediante herramientas gráficas como MySQL Workbench, DBeaver o TablePlus, conectadas al contenedor Docker a través del puerto 3306.

Ejemplo de conexión vía terminal:

```
PS C:\Users\stine> docker exec -it mysql-chuok mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.42 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

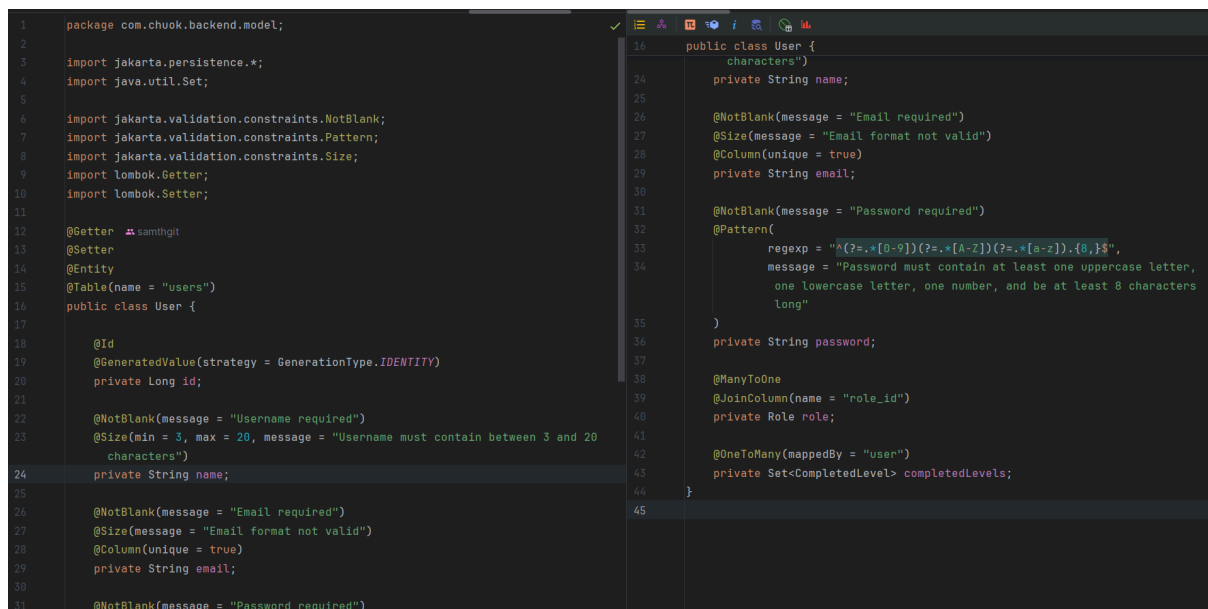
En cuanto a las tablas, hay que destacar que la base de datos de la aplicación no ha sido definida directamente mediante scripts SQL, sino que ha sido generada de forma automática a través del uso de JPA (Java Persistence API) junto con Hibernate, el motor ORM que se encarga de traducir los modelos de datos de Java en tablas relacionales en MySQL.

ORM (Object-Relational Mapping) es una técnica de programación que permite convertir datos entre sistemas incompatibles. Gracias a ello, los desarrolladores pueden trabajar con objetos y sus relaciones en el código, mientras que el sistema se encarga de traducir automáticamente estas estructuras a tablas, filas y columnas en la base de datos. Esto simplifica el desarrollo, reduce errores y mejora la mantenibilidad del código.

Cada entidad del dominio (como User, Role, Article, Phrase, etc.) ha sido mapeada a una tabla de base de datos mediante anotaciones como `@Entity`, `@Table`, `@Id`, `@ManyToOne`, etc. Este enfoque permite mantener el modelo de datos sincronizado con el código, facilitando su evolución y mantenimiento.

Por ejemplo, la entidad User representa a un usuario del sistema, con atributos como nombre, correo electrónico y contraseña, y relaciones con otras entidades como Role y CompletedLevel. Al ejecutar la aplicación, Spring Boot se encarga de detectar estas clases y generar automáticamente las tablas correspondientes en la base de datos, utilizando la configuración establecida en el archivo `application.properties`.

A continuación se muestra el código con la estructura de la entidad:



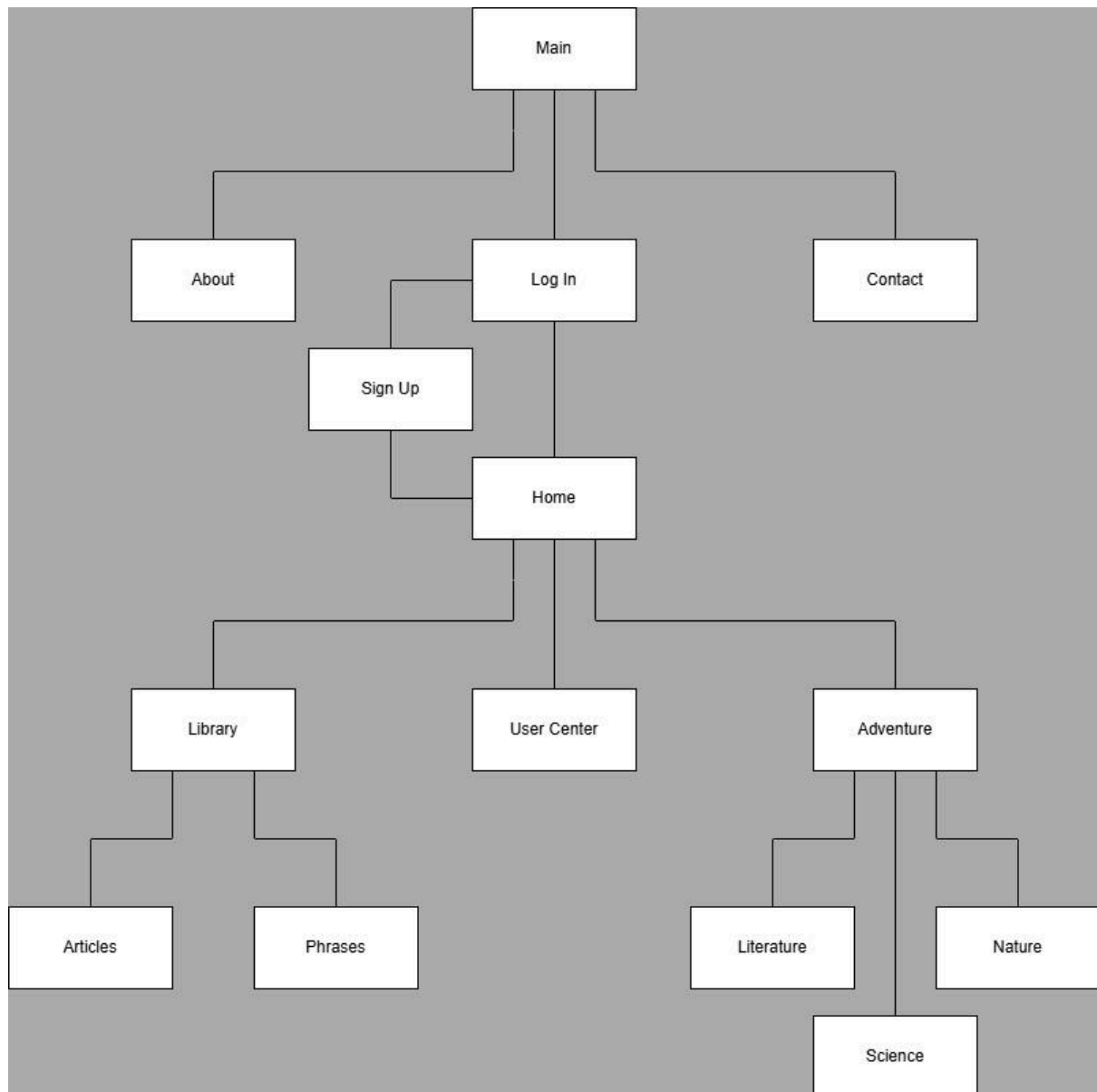
```
1 package com.chuok.backend.model;
2
3 import jakarta.persistence.*;
4 import java.util.Set;
5
6 import jakarta.validation.constraints.NotBlank;
7 import jakarta.validation.constraints.Pattern;
8 import jakarta.validation.constraints.Size;
9 import lombok.Getter;
10 import lombok.Setter;
11
12 @Getter
13 @Setter
14 @Entity
15 @Table(name = "users")
16 public class User {
17
18     @Id
19     @GeneratedValue(strategy = GenerationType.IDENTITY)
20     private Long id;
21
22     @NotBlank(message = "Username required")
23     @Size(min = 3, max = 20, message = "Username must contain between 3 and 20 characters")
24     private String name;
25
26     @NotBlank(message = "Email required")
27     @Size(message = "Email format not valid")
28     @Column(unique = true)
29     private String email;
30
31     @NotBlank(message = "Password required")
32     @Pattern(
33         regexp = "^(?=.*[0-9])(?=.*[A-Z])(?=.*[a-z]).{8}$",
34         message = "Password must contain at least one uppercase letter, one lowercase letter, one number, and be at least 8 characters long"
35     )
36     private String password;
37
38     @ManyToOne
39     @JoinColumn(name = "role_id")
40     private Role role;
41
42     @OneToMany(mappedBy = "user")
43     private Set<CompletedLevel> completedLevels;
44 }
45
```

De esta manera, al ejecutar la aplicación, podemos ir a nuestra base de datos y comprobar que la tabla ha sido creada correctamente:


```
mysql> DESCRIBE users;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | bigint        | NO   | PRI | NULL    | auto_increment |
| email | varchar(255)  | YES  | UNI | NULL    |                |
| name  | varchar(20)   | NO   |     | NULL    |                |
| password | varchar(255) | YES  |     | NULL    |                |
| role_id | bigint        | YES  | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Este enfoque basado en código es especialmente útil durante el desarrollo, ya que evita errores de sincronización entre el modelo y la estructura real de la base de datos y acelera el proceso de prototipado.

- c. Diseño del sitio web:
 - i. Mapa del sitio web, indicando las páginas existentes y los enlaces entre estas



A continuación, una explicación exhaustiva de cada página, incluyendo sus elementos y enlaces con otras páginas:

- **MAIN**
Página principal a la que accede el usuario que no ha sido previamente registrado.
Página donde se redirige al usuario cuando accede a una página que no existe.
Contenido:
 - Bloque de contenido que resume las características principales de la aplicación.
 - Reviews de usuarios de chuOK.
- **ABOUT**
Página dedicada a informar al usuario sobre las funcionalidades e intrahistoria de la aplicación.

- Contenido:
 - Contenedor dedicado a descripción e historia de chuOK.
 - Contenedor dedicado al perfil del creador.
- CONTACT

Página que permite al usuario entrar en contacto con el creador.

Contenido:

 - Formulario con datos de contacto del usuario.
- LOG IN

Página de inicio de sesión. Página donde se redirige al usuario al registrarse o cerrar sesión.

Contenido:

 - Formulario de inicio de sesión consistente en email + contraseña.
- SIGN UP

Página de registro.

Contenido:

 - Formulario de registro consistente en nombre de usuario, email y contraseña.
- HOME

Página principal del usuario ya registrado, donde tiene acceso a las diferentes secciones de la aplicación.

Contenido:

 - Título a modo de saludo con nombre personalizado (el del usuario).
 - Contenedor que da acceso a la biblioteca y al modo aventura.
- LIBRARY

Página principal de la sección libre, da acceso a las secciones específicas.

Contenido:

 - Título y subtítulo.
 - Contenedor que enlaza con las páginas de Artículos y Frases.
- ARTICLES

Página encargada de mostrar artículos al usuario. Se encuentran dispuestos a modo de rejilla, ordenados por fecha, de manera que los últimos artículos subidos aparecen primero.

Contenido:

 - Barra de navegación especial que comunica con la sección de Frases.
 - Grid de objetos ARTICLE.

- PHRASES

Página encargada de mostrar frases célebres al usuario. Se encuentran dispuestas a modo de rejilla, ordenadas por fecha, de manera que las últimas frases subidas aparecen primero.

Contenido:

- Barra de navegación especial que comunica con la sección de Artículos.
- Grid de objetos PHRASE.

- ADVENTURE

Página principal de la sección de modo aventura. Da acceso a los diferentes mundos.

Contenido:

- Título y subtítulo.
- Contenedor que enlaza con los mundos (páginas) de Literatura, Ciencia y Naturaleza.

- LITERATURE

Primer mundo del modo aventura. Los niveles se encuentran dispuestos a modo de camino donde el usuario tiene que hacer scroll para verlos todos.

Contenido:

- Barra de navegación especial que comunica con los otros mundos (Science y Nature).
- Contenedor de altura superior a la de la ventana que contiene los niveles, accesibles a través de un botón.

- SCIENCE

Segundo mundo del modo aventura. Los niveles se encuentran dispuestos a modo de camino donde el usuario tiene que hacer scroll para verlos todos.

Contenido:

- Barra de navegación especial que comunica con los otros mundos (Literature y Nature).
- Contenedor de altura superior a la de la ventana que contiene los niveles, accesibles a través de un botón.

- NATURE

Segundo mundo del modo aventura. Los niveles se encuentran dispuestos a modo de camino donde el usuario tiene que hacer scroll para verlos todos.

Contenido:

- Barra de navegación especial que comunica con los otros mundos (Literature y Science).

- Contenedor de altura superior a la de la ventana que contiene los niveles, accesibles a través de un botón.
- USER CENTER
 Página que muestra datos generales que el usuario puede rellenar para dar más información personal.
 Contenido:
 - Contenedor con barra de navegación con tres secciones.
 - Profile: El usuario puede dar más datos personales, como el nombre completo, país y biografía.
 - Account: El usuario tiene acceso al nombre de usuario, email y contraseña con los que cuenta dentro de la aplicación.
 - Settings: Apartado general donde de momento el usuario puede cambiar el idioma de la interfaz.

Para la ejecución del mapa del sitio web se ha hecho uso de la aplicación Draw.io.

Draw.io es una herramienta gratuita de diagramación en línea, muy utilizada para crear diagramas de flujo, mapas conceptuales, mapas de sitio web, diagramas UML, modelos entidad-relación, etc.

Funciona directamente en el navegador, sin necesidad de instalar nada.

Más información en el Anexo 3

5. Proceso de creación

En este apartado vamos a ir hasta el fondo de la creación de chuOK. Discutiremos sobre las tecnologías usadas y un walkthrough del proceso de creación de los elementos más notorios del proyecto. De igual manera que chuOK, vamos a dividir esta sección en dos; front-end, donde tiene lugar la parte visual de la aplicación gracias al framework Angular, y back-end, en el que con la ayuda del framework Spring Boot se tratan otros aspectos de la aplicación como REST API, conexión a base de datos, entre otros.

De esta manera comenzamos con el front-end. Dada la temática de la aplicación, indagué sobre la psicología del color y cómo afecta a las personas. Finalmente me decidí por el verde como color principal de la aplicación (se usa tanto en el logo como en las páginas web) y para el color secundario me decidí por el color complementario, contrario en el espectro de colores, el morado. El blanco y el negro son partícipes para dar contrastes, eso sí, dada la naturaleza del proyecto, me decidí por variantes un poco más suaves de estos colores, un gris muy claro (#f5f5f5) sustituyó al blanco puro y el gris oscuro suave (#292929) hizo lo propio con el negro puro.

¿Cómo afecta el color al cerebro humano?

Los colores estimulan diferentes regiones del cerebro. Cuando vemos un color:

El ojo lo percibe como una longitud de onda específica.

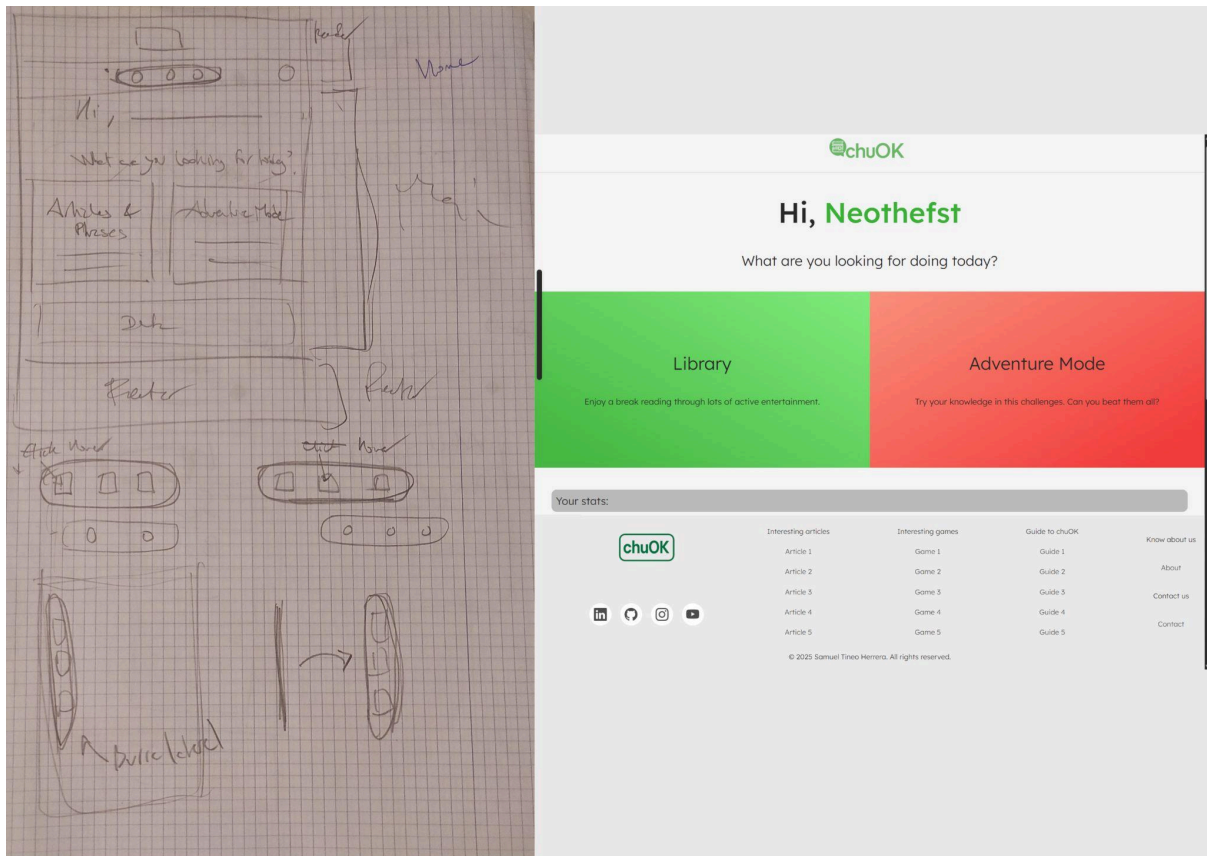
Esa información se transmite al hipotálamo, que regula respuestas fisiológicas (como la temperatura corporal, el apetito, y el estado de ánimo).

Se desencadenan respuestas emocionales y cognitivas.

Por ejemplo, los colores cálidos pueden aumentar la frecuencia cardíaca, mientras que los fríos tienden a relajar.

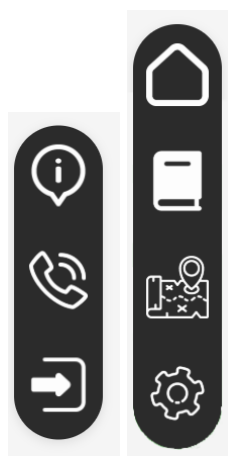
Más información en el Anexo 2

En lo que respecta a las vistas lo tenía claro: claridad. Un espacio libre de distracciones donde todos los elementos estuvieran espaciados, eliminando esa sensación de agobio que puede producir en el usuario el tener demasiado puntos visuales donde enfocar su atención. Así nació la idea del menú desplegable, la barra de navegación se sitúa a un lado y aunque siempre se mantiene en pantalla, no ofrece ningún ruido visual ya que en su estado natural el usuario ve una simple barra.

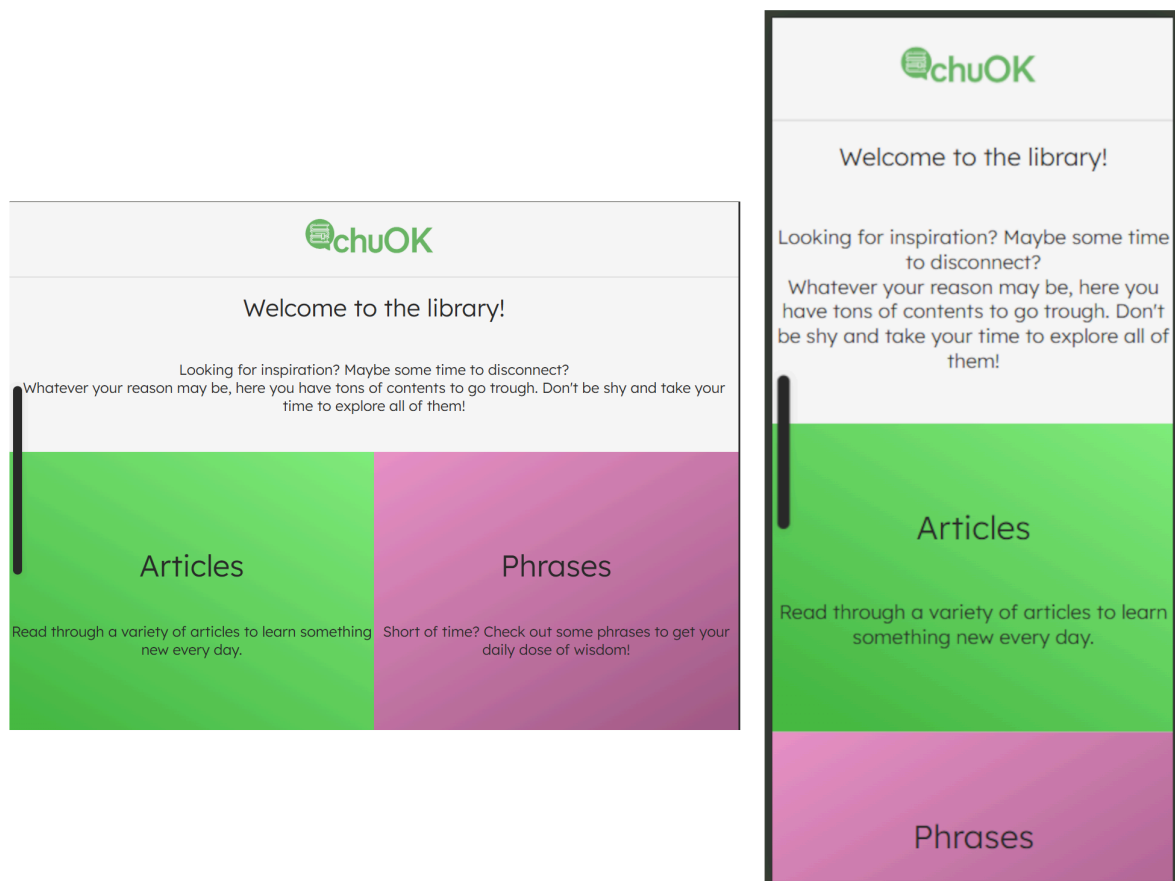


En esta imagen podemos ver la idea original y las transformaciones que sufrió en el propio boceto a la izquierda, y en el lado opuesto podemos ver la view terminada, incorporando ya el menú lateral desplegable.

Otro factor que tuve en cuenta para seguir el patrón de la claridad fue la sustitución del texto por logos, todos los accesos de los menús desplegables se muestran al usuario como logos, para que este asocie la imagen con el destino al que le lleva y no sea simplemente texto plano, un pequeño gesto que reduce la monotonía y da un aspecto refrescante al sitio web.



Para acabar este apartado de introducción al proceso de creación me gustaría destacar la responsividad de cada una de las vistas que componen la aplicación, que han sido cuidadosamente diseñadas para hacer la experiencia del usuario de dispositivo móvil lo más fluida e interesante posible. A continuación se muestra un ejemplo de una vista en su forma de boceto, diseño para ordenador y diseño para móvil. *Consultar el Anexo 5 para ver más ejemplos.*



FRONT-END

En este apartado vamos a tratar sobre los aspectos de chuOK concernientes al front-end. Lo primero, como es natural, es conocer la herramienta que hace posible llevar a cabo esta tarea. Tras un estudio de diferentes frameworks, me acabé decantando por Angular, ya que, entre otras cosas, es el framework de front-end tratado en clase.

Angular es un framework de desarrollo web de código abierto, mantenido principalmente por Google, que permite crear aplicaciones web SPA (Single Page Applications) con una arquitectura robusta y escalable.

Se basa en TypeScript, un superconjunto de JavaScript que añade tipado estático y otras características avanzadas.

Más información en el Anexo 6

¿Por qué Angular?

chuOK presenta una serie de características dinámicas, personalizadas para cada usuario y múltiples componentes reutilizables. De esta manera, Angular es la elección apropiada:

Arquitectura basada en componentes

Angular permite dividir la UI en componentes independientes y reutilizables (por ejemplo, login, registro, frase, artículo, nivel, mundo).

Esto es perfecto para chuOK, donde cada sección de la app tiene diferentes vistas (según tipo de usuario, mundo desbloqueado, etc.).

UI (User Interface) – Interfaz de Usuario

Es el diseño visual y la disposición de los elementos con los que el usuario interactúa: botones, menús, tipografía, colores, iconos, etc. Se enfoca en cómo se ve y se maneja una aplicación.

Routing avanzado

Con el router de Angular puedes manejar navegación entre mundos, niveles, zonas bloqueadas, etc., sin recargar la página.

También se puede proteger rutas según el rol del usuario (normal, admin), lo cual es clave para gestionar accesos diferenciados.

Reactive Forms y Validaciones

Angular tiene herramientas avanzadas para formularios reactivos, perfectas para la pantalla de registro e inicio de sesión, con validaciones complejas del lado del cliente.

Servicios e inyección de dependencias

Se puede centralizar la lógica de comunicación con el backend (Spring Boot) en servicios Angular bien estructurados y testables.

Manejo del estado del usuario

Es posible usar guards, interceptors y servicios compartidos para manejar el estado del usuario, el token JWT, los privilegios y otras reglas de acceso a vistas.

Experiencia de usuario fluida (SPA)

Al ser una SPA (Single Page Application), Angular evita recargas completas y proporciona una experiencia (UX) más fluida y moderna.

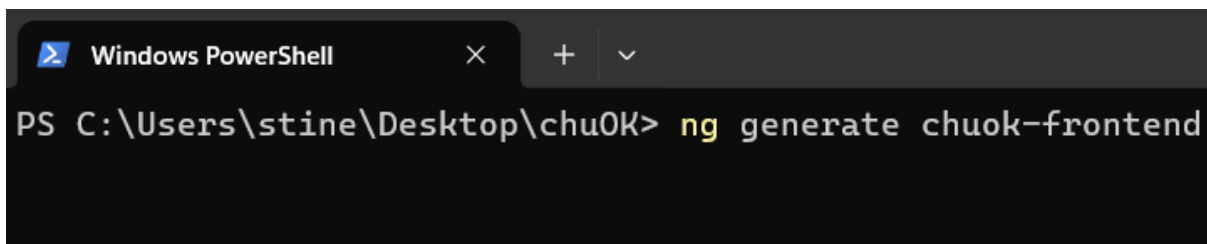
UX (User Experience) – Experiencia de Usuario

Es la percepción general del usuario al interactuar con un producto o servicio. Involucra la facilidad de uso, la eficiencia, la satisfacción y la lógica de navegación.
Se enfoca en cómo se siente y funciona la experiencia.

Creación del proyecto

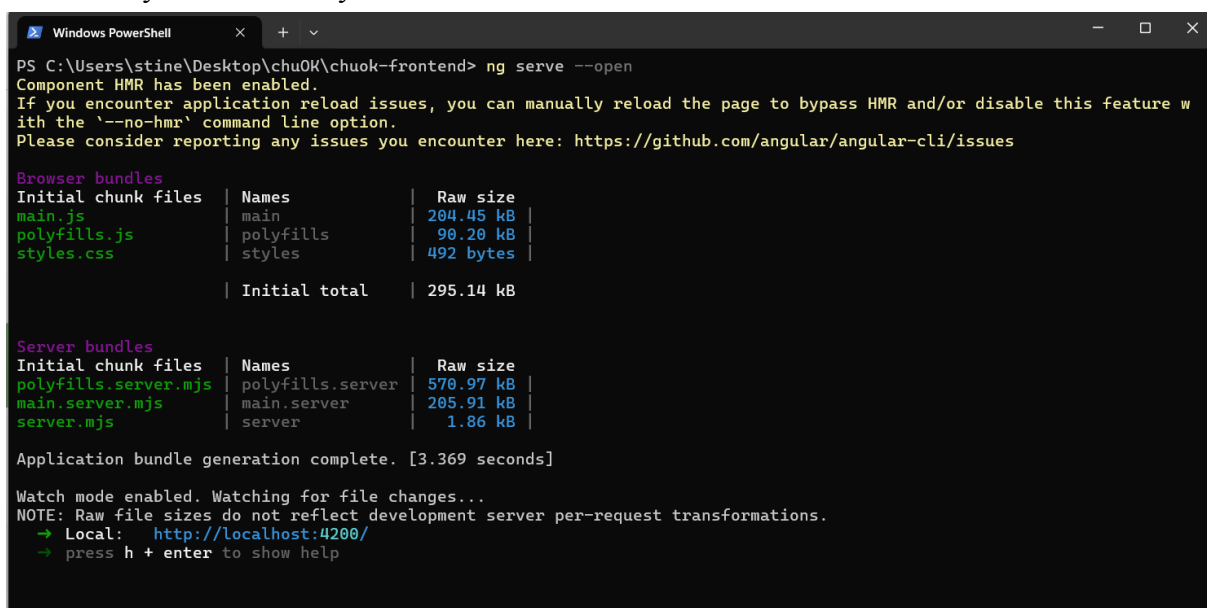
A continuación se muestran una serie de capturas de pantalla que ilustran de manera general algunos comandos necesarios para la creación y posterior visualización del proyecto. Todos ellos tienen el común denominador 'ng', que denota que es un comando propio de Angular CLI

Empezamos creando el proyecto usando el comando ng generate nombre-proyecto



```
Windows PowerShell
PS C:\Users\stine\Desktop\chuOK> ng generate chuok-frontend
```

Una vez terminado el proceso, vamos a comprobar que la instalación se ha efectuado con éxito con el comando ng serve --open, que nos abrirá una ventana en nuestro navegador con el servicio ya inicializado y corriendo



```
Windows PowerShell
PS C:\Users\stine\Desktop\chuOK\chuok-frontend> ng serve --open
Component HMR has been enabled.
If you encounter application reload issues, you can manually reload the page to bypass HMR and/or disable this feature with the '--no-hmr' command line option.
Please consider reporting any issues you encounter here: https://github.com/angular/angular-cli/issues

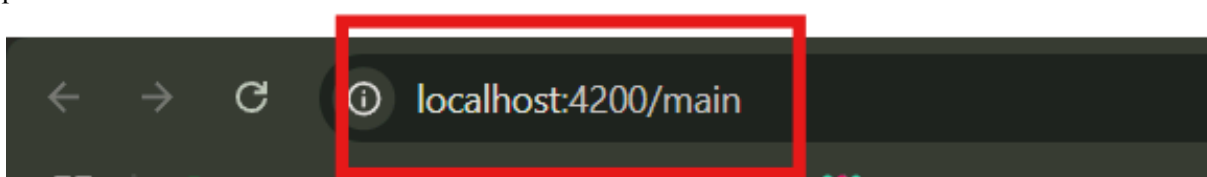
Browser bundles
Initial chunk files | Names | Raw size
main.js | main | 204.45 kB
polyfills.js | polyfills | 90.20 kB
styles.css | styles | 492 bytes
Initial total | 295.14 kB

Server bundles
Initial chunk files | Names | Raw size
polyfills.server.mjs | polyfills.server | 570.97 kB
main.server.mjs | main.server | 205.91 kB
server.mjs | server | 1.86 kB

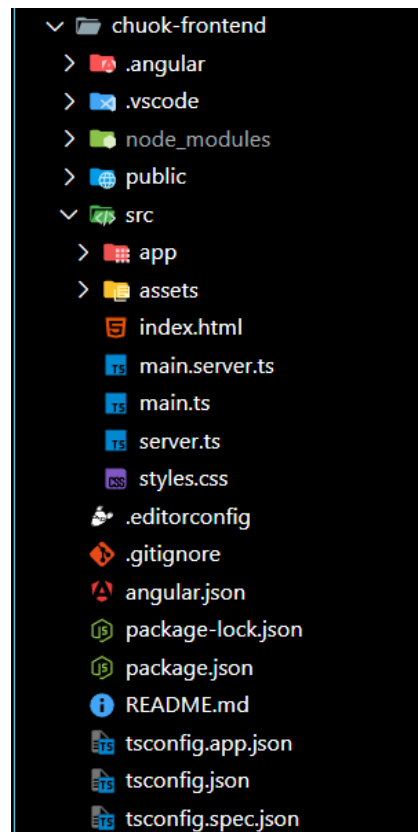
Application bundle generation complete. [3.369 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
→ Local: http://localhost:4200/
→ press h + enter to show help
```

Cuando el servicio se ejecuta, como se puede apreciar en la anterior imagen, Angular usa como puerto por defecto el 4200

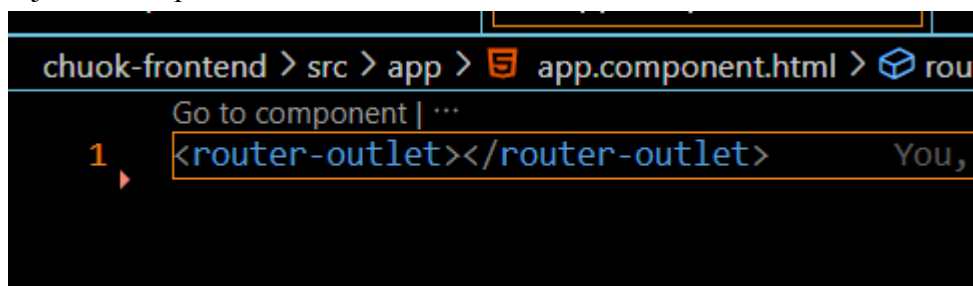


Si ahora nos vamos a nuestro IDE deberíamos ver una estructura similar a esta:



No vamos a hacer un tutorial completo del funcionamiento de Angular, pero sí nos vamos a parar en algunos de los detalles más esenciales de este framework.

Lo primero que hay que hacer es localizar, el app-component, el componente raíz de nuestro proyecto, y por el que todos nuestros componentes deben pasar si quieren ser renderizados en la página web. Luego nos detendremos a mirar más en detalle la estructura de los componentes, pero por ahora, solo hay que saber que para que estos se muestren es necesario dejar esta etiqueta:



12. Bibliografía

ANEXO 1:

[83% of developers suffer from burnout | IT Pro](#)
[Burn-out an occupational phenomenon](#)

[World Health Organisation widens definition of burnout | Prospect](#)

<https://www.genbeta.com/a-fondo/desarrolladores-estan-quemadisimos-mayoria-se-divierte-programando-fines-semana>

13. Anexos

ANEXO 1: Burnout

El burnout o síndrome de agotamiento profesional es un trastorno psicológico relacionado con el entorno laboral que se caracteriza por un estado de estrés crónico, agotamiento emocional y despersonalización. Se reconoció oficialmente como un fenómeno laboral por la Organización Mundial de la Salud (OMS) en 2019 y está clasificado en la CIE-11 (Clasificación Internacional de Enfermedades) como un fenómeno ocupacional, no como una condición médica.

El burnout se desarrolla como respuesta prolongada a factores de estrés emocional e interpersonal en el trabajo. Se manifiesta en tres dimensiones principales:

- Agotamiento emocional
 - Sensación de estar exhausto física y mentalmente.
 - Falta de energía para enfrentar la jornada laboral.
 - Sentimientos de desesperanza o de que no se puede seguir adelante.
- Despersonalización o cinismo
 - Actitudes negativas o distantes hacia el trabajo y hacia otras personas (compañeros o clientes).
 - Pérdida de empatía y desconexión emocional.
 - Trato impersonal, incluso hostil.
- Baja realización personal
 - Sentimiento de ineficacia o de no estar logrando objetivos.
 - Pérdida de motivación.
 - Sensación de inutilidad o fracaso profesional.

Causas comunes:

- Sobrecarga de trabajo constante.
- Falta de control sobre el propio trabajo o sobre decisiones importantes.

- Expectativas poco claras o cambiantes.
- Falta de reconocimiento o recompensas.
- Ambiente laboral tóxico (malas relaciones, acoso, falta de apoyo).
- Desajuste entre los valores personales y los de la empresa.
- Pobre equilibrio entre la vida personal y profesional.

Consecuencias físicas y psicológicas:

El burnout no solo afecta el rendimiento laboral, también puede tener efectos profundos en la salud física y mental:

Consecuencias psicológicas:

- Ansiedad
- Depresión
- Insomnio
- Irritabilidad o cambios de humor
- Aislamiento social
- Dificultad para concentrarse

Consecuencias físicas:

- Fatiga crónica
- Dolores musculares y de cabeza
- Problemas gastrointestinales
- Problemas cardiovasculares (hipertensión, taquicardia)
- Mayor vulnerabilidad a enfermedades

¿Cómo se puede prevenir o tratar?

A nivel personal:

- Establecer límites claros entre vida laboral y personal.
- Dormir y comer adecuadamente.
- Practicar ejercicio y técnicas de relajación (como mindfulness).
- Buscar apoyo emocional (familia, amigos, terapia).
- Aprender a delegar y decir “no”.

A nivel organizacional:

- Fomentar un ambiente laboral sano.
- Capacitar a líderes y supervisores para detectar señales de agotamiento.
- Dar autonomía y claridad en las tareas.
- Ofrecer programas de bienestar y salud mental.
- Reconocer y recompensar el buen trabajo.

¿En qué se diferencia del estrés?

El estrés es una respuesta puntual al exceso de presión y suele disminuir cuando el factor estresante desaparece.

El burnout, en cambio, es un estado crónico que se desarrolla a lo largo del tiempo y no se alivia fácilmente, incluso si se eliminan los factores estresantes.

Burnout en el sector del desarrollo de software:

Datos sobre el burnout en desarrolladores de software

- Prevalencia: Según el informe State of Developer Ecosystem 2023 de JetBrains, el 73% de los desarrolladores han experimentado burnout en algún momento de su carrera.
- Factores contribuyentes: Un estudio de Haystack Analytics identificó las principales causas del burnout entre desarrolladores: alta carga de trabajo (47%), procesos ineficientes (31%) y objetivos poco claros (29%).

- Impacto de la pandemia: El mismo estudio de Haystack Analytics señaló que el 81% de los desarrolladores reportaron un aumento en el burnout debido a la pandemia, siendo el incremento de la carga laboral la principal razón.