

PROYECTO DAW

LinguoLearn



ÍNDICE

DESCRIPCION GENERAL	1
IDEA GENERAL	1
FUNCIONALIDAD PRINCIPAL	3
PROPOSITOS Y OBJETIVOS	3
ALCANCE DEL SISTEMA	3
FUNCIONALIDAD	3
TIPOLOGIA DE USUARIOS	4
Modelo Entidad Relación	5
MAQUETACION	6
DISEÑO DE MARCA	12
RESULTADOS	13
Despliegue	25
Analitica web, posicionamiento y promoción	25
Planificación temporal	26
Plazo de entrega	26
Garantia y soporte	26
Licencia v propiedad intelectual	27

DESCRIPCION GENERAL

LinguoLearn es una innovadora aplicación web educativa diseñada para facilitar el aprendizaje de idiomas de manera efectiva y entretenida. Nuestra plataforma se centra en ofrecer a los usuarios una experiencia de aprendizaje de idiomas similar a un juego, combinada con un enfoque pedagógico sólido. El objetivo principal de LinguoLearn es empoderar a las personas de todas las edades y niveles de habilidad para que dominen nuevos idiomas, ya sea por razones personales, profesionales o académicas.

IDEA GENERAL

La idea central de LinguoLearn es proporcionar una herramienta integral para el aprendizaje de idiomas que sea accesible desde cualquier dispositivo con conexión a internet. Los usuarios pueden elegir entre una amplia variedad de idiomas, incluyendo los más hablados en el mundo, y avanzar a su propio ritmo a

través de lecciones interactivas y divertidas. LinguoLearn se basa en el aprendizaje mediante la practica con pequeños tests interactivos.

FUNCIONALIDAD PRINCIPAL

- Lecciones interactivas: LinguoLearn ofrece lecciones estructuradas que cubren vocabulario, gramática, expresiones cotidianas y habilidades de conversación en el idioma elegido.
- Evaluación de habilidades: Los usuarios pueden realizar pruebas de evaluación para medir su progreso y determinar su nivel de competencia en el idioma.
- Experiencia de juego: La aplicación incorpora elementos de gamificación, como recompensas y desafíos, para mantener a los usuarios motivados y comprometidos.

PROPOSITOS Y OBJETIVOS

Los propósitos y objetivos de LinguoLearn son los siguientes:

- 1. Facilitar el aprendizaje de idiomas de manera efectiva y accesible.
- 2. Ayudar a los usuarios a alcanzar la fluidez en un nuevo idioma.
- 3. Fomentar la diversidad lingüística y cultural.
- 4. Apoyar metas personales, profesionales y académicas de los usuarios.
- 5. Proporcionar una experiencia de aprendizaje atractiva y motivadora.

En resumen, LinguoLearn es la solución definitiva para aquellos que desean aprender un nuevo idioma de manera efectiva y divertida, aprovechando la tecnología y principios pedagógicos de vanguardia.

ALCANCE DEL SISTEMA FUNCIONALIDAD

Usuario:

- Registro de nuevos usuarios
- Inicio de sesión para usuarios existentes

Idioma:

Un usuario podrá elegir uno o varios idiomas

Prueba de nivel:

- Posibilidad de realizar una prueba de nivel al inicio para determinar tu nivel
- Posibilidad de ir realizando pruebas de nivel para avanzar de nivel
- Registro de resultados de las pruebas realizadas por el usuario

Nivel:

- Almacenamiento del nivel actual de cada usuario en un idioma especifico
- Actualización del nivel a medida que el usuario progresa

Ejercicios:

- Acceso a ejercicios de tipo test/escribir/arrastrar palabras para formar frases
- Ejercicios relacionados con el idioma y el tipo (vocabulario, gramática, etc)
- Avance gradual a través de los ejercicios
- Registro de progreso del usuario en los ejercicios
- Retroalimentación sobre los errores del usuario en los ejercicios
- Acceso a prueba de nivel tras finalizar los ejercicios del nivel actual del usuario.

TIPOLOGIA DE USUARIOS

Usuario anónimo:

- Registrarse o iniciar sesión en la aplicación.
- Ver la pantalla de login

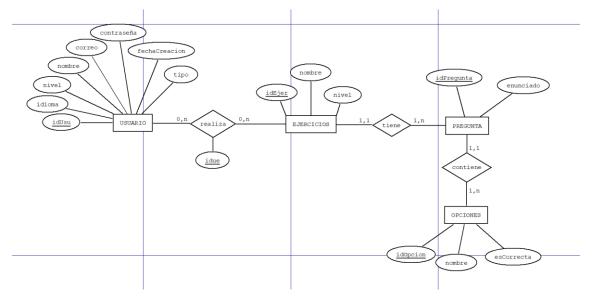
Usuario registrado:

- Iniciar sesión en la aplicación
- Editar información de su perfil
- Consultar su perfil
- Ver el listado de idiomas
- Hacer tests de su idioma
- Hacer pruebas de nivel para cada nivel de su idioma

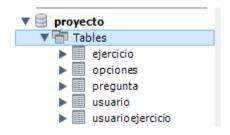
Administrador:

Podrá añadir, modificar o eliminar ejercicios, preguntas, opciones y usuarios

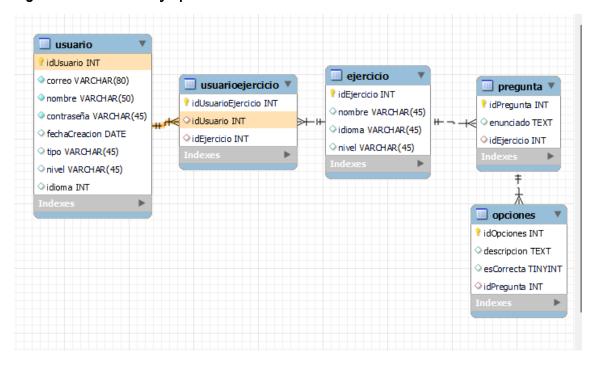
Modelo Entidad Relación



Tablas en MySql Workbench:



Ingeniería inversa en MySql Workbench:

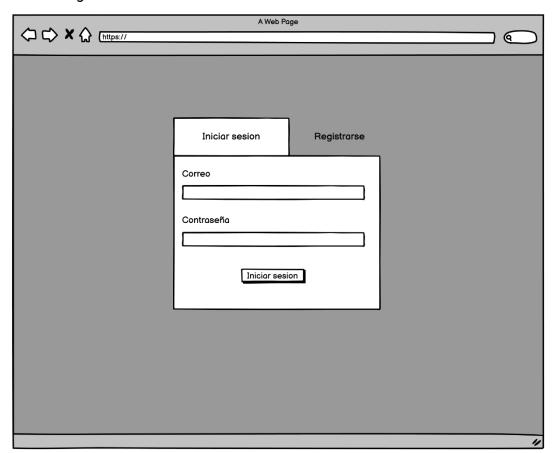


MAQUETACION

Nuevo usuario:

A Web Page A Web Page A Web Page				
₩ (maps.//				
	Iniciar sesion	Reigstrarse		
	Usuario:			
	Соггео			
	Contraseña			
	Regist	rarse		
			,	
				1/

Usuario registrado:



Nuevo usuario:



Nuevo usuario – pregunta prueba de nivel:



Nuevo usuario – prueba de nivel:



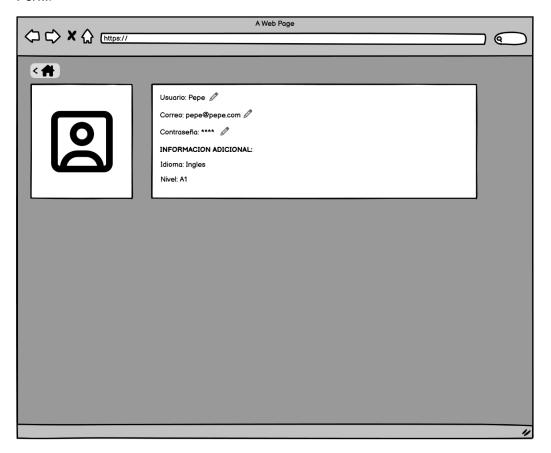
Nuevo usuario - fin prueba de nivel:



Main page:



Perfil:



ADMINISTRADOR:



Al hacer click sobre la pregunta que deseas editar:



Se podrá editar, eliminar o convertir en verdadera cada opción de la pregunta además de editar la pregunta.

DISEÑO DE MARCA

Naming

El nombre de la marca nace del anglicismo de la palabra "Aprender" en inglés "Learn" y la palabra lenguaje "Lenguaje" en latín "Lingua" que unidas forman LinguoLearn

Logotipo e isotipo

La idea del logotipo se basa en la simpleza y la idea de atraer a todo tipo de públicos a través de crear una imagen corporativa agradable y sencilla. Por ese motivo se utilizara el propio nombre de la marca con una fuente simple y firme junto con la presencia de un gorrión para conseguir esa parte de amigabilidad



Como isotipo identificativo de la marca se utilizara el gorrión ya que es una manera rápida y simple de identificar la marca



Tipografias y colores corporativos

La tipografía busca transmitir simpleza y minimalismo a la experiencia de usuario por lo tanto se decide utilizar una única tipografía "**Poppins**"

Regular - 400

Whereas recognition of the inherent dignity

Bold - 700

Whereas recognition of the inherent dignity

Los colores se centran en el azul buscando una paleta de azules concordante con el azul del logo y buscando distintas tonalidades para poder jugar con ellas en la aplicación, finalmente los colores principales son:

#1899D6 - Azul énfasis (botones)

#DOEBF5 - Azul fondo

#333 - Texto

#2fc4ff85 - Azul principal

#32CD32 - Verde aciertos

#ff3333 - Rojo error

RESULTADOS

Usuario anonimo:

Pagina login – Inicio de sesión

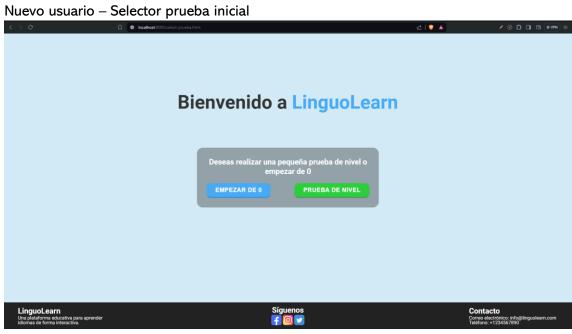




Nuevo usuario:

Nuevo usuario - Selector de idioma









Usuario registrado:

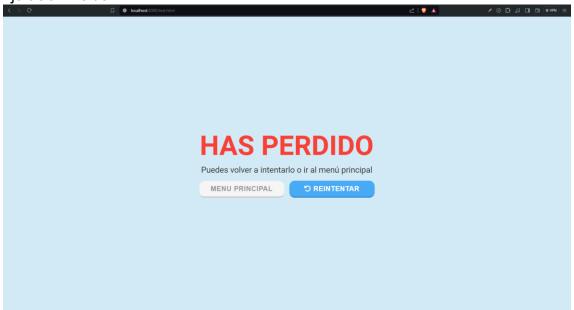
Main page



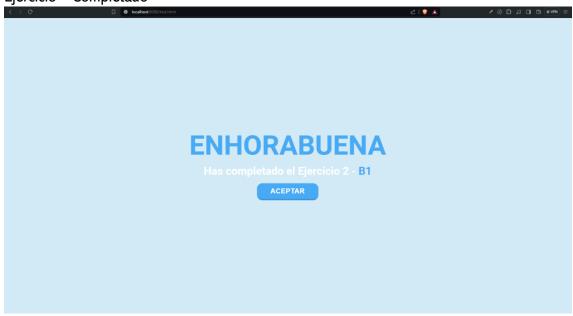
Ejercicio:



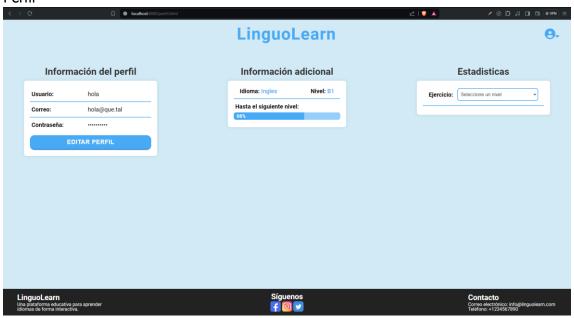
Ejercicio – Perder



Ejercicio – Completado



Perfil



Peril - Editar perfil

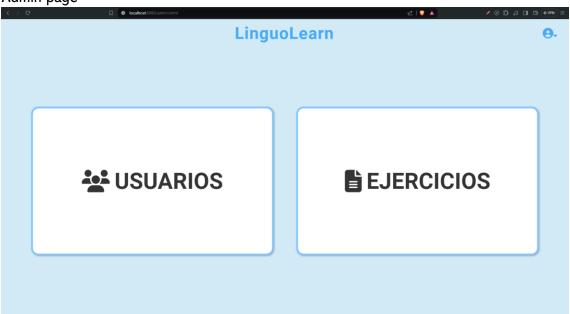


Perfil - Estadísticas



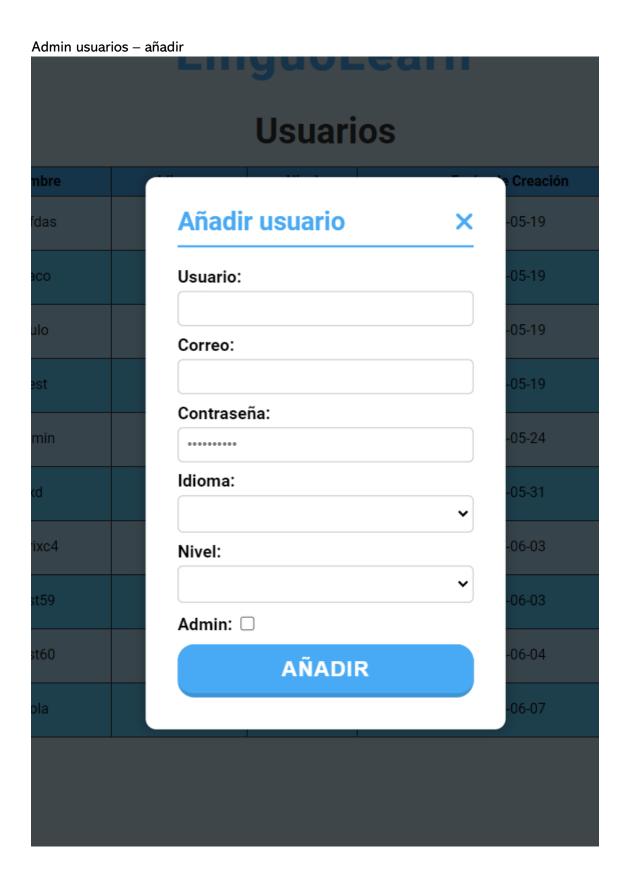
Admin:

Admin page



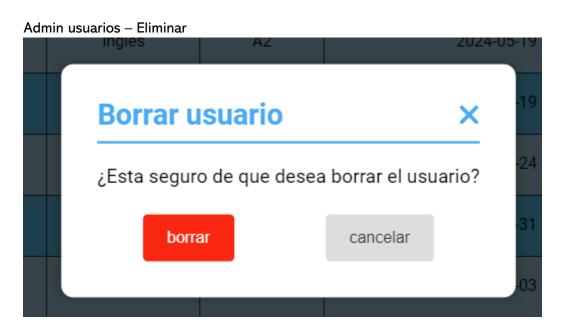
Admin usuarios





Admin usuarios - Editar





Admin ejercicios



Admin preguntas



Admin opciones



Despliegue

Dominio

El dominio elegido será www.linguolearn.es



Alojamiento

La aplicación se alojará de manera gratuita utilizando el servicio de Railway que nos permite desplegar una aplicación de **Spring** con un crédito inicial de 5\$ y un servidor de 512 MB de ram, 1GB de disco duro y 2 vCPU (La aplicación con estas limitaciones sufre bastante lentitud)

Aplicación desplegada: https://linguolearn.up.railway.app/

Analitica web, posicionamiento y promoción

Analítica web

Una vez realizado el despliegue definitivo, se planea realizar a través de **Google Analytics**, ya que es **gratuito** y tiene una gran cantidad de funcionalidades.

Esto permitirá tomar decisiones en función de los datos recogidos durante el análisis y tener claro al **público objetivo** al que estamos llegando con las diferentes **promociones y campañas publicitarias**.

Posicionamiento

El posicionamiento **SEO** se realizará a través de diferentes técnicas y características aplicadas a la estructura de la aplicación web, para que el **indexado** de los **motores de búsquedas** la mantengan accesible y fácilmente **localizable**.

Promoción

Los principales canales donde promocionar nuestra aplicación son las redes sociales, ya que es donde más publico podemos llegar.

A través de campañas y promociones en nuestras cuentas oficiales como **Twitter**, **Instagram y Facebook** se dará **visibilidad y reconocimiento** a nuestra aplicación.

Planificación temporal

	Fecha inicio	Fecha fin
Planificación	25/09/2023	03/10/2023
Back-end		
Estructura inicial	3/10/2023	24/10/2023
Modelo de datos	10/11/2023	1/12/2023
Repositorios y servicios	1/12/2023	1/1/2024
Controladores	1/1/2024	10/02/2024
Front-end		
Login	10/02/2024	1/03/2024
Test	1/03/2024	1/04/2024
Perfil	1/04/2024	25/04/2024
Panel de administrador	1/05/2024	1/06/2024
Documentacion	1/06/2024	7/06/2024

Plazo de entrega

La entrega del proyecto está fechada para el 08/06/2024.

Garantia y soporte

Garantia

El periodo de garantía será de 12 meses en los que se garantizará:

- Funcionamiento: La aplicación web funcionara de acuerdo con las especificaciones definidas en este documento
- Corrección de errores: Se solucionarán errores que afecten en el rendimiento y funcionalidad de la aplicación
- Actualizaciones: Se realizarán actualizaciones periódicas para corregir errores y/o añadir contenido como ejercicios, preguntas, idiomas, etc...

Soporte

El soporte durara también 12 meses, en los que se garantiza:

- **Soporte técnico:** Se dará un soporte al cliente y usuario para resolver cualquier tipo de problema, consulta o solicitud de ayuda.
- **Tiempo de respuesta:** Se atenderán todas las solicitudes de soporte en un plazo inferior a 24 horas.
- Copias de seguridad: Se harán copias de respaldo con toda la información de la base de datos y código de la aplicación semanalmente.

Licencia y propiedad intelectual

Licencia

Tanto la marca LinguoLearn como la aplicación web cuentan con licencia de Copyright.

- El autor asegura el reconocimiento de su obra
- Es el beneficiario de las gestiones comerciales de esta
- Evita en todo momento que alguien sea capaz de robarle su autoría
- Puede decidir la accesibilidad de su obra en cualquier momento. Desde la opción de todos los derechos reservados que lo mas habitual es que se comparta haciéndose referencia a su autoría.

Propiedad intelectual

La marca de **LinguoLearn** pertenece al titular de la aplicación, el desarrollador en este caso.

Las imágenes, preguntas y diferente información de la aplicación es propiedad de la marca.

Forma de pago

El pago se realizará en 3 fases:

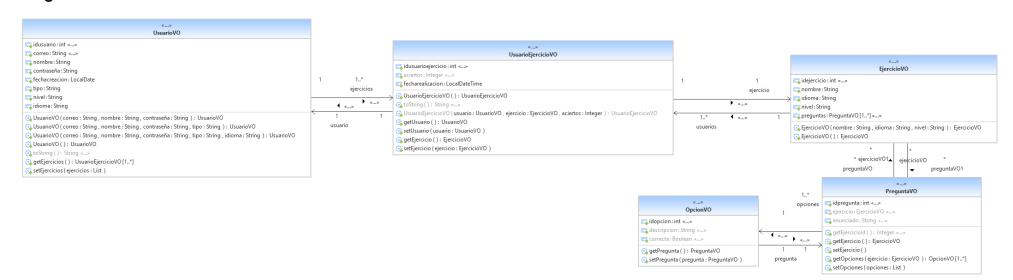
- 20% a modo de señal, tras la firma del funcional y presupuesto.
- 40% a la mitad de la etapa de desarrollo
- 40% tras despliegue y entrega de la aplicación

La aplicación esta valorada en unas **500 horas** en total las cuales se cobrarán a 15€ la hora (precio/hora obtenido de https://www.calculadorafreelance.com/)

Este pago no incluye el soporte y mantenimiento de la aplicación el cual se valorará con el cliente tras la entrega de la aplicación.

Diagramas

Diagrama de clases - VO



Repositorios



Servicios

Gflush(): Gflush(): Gflush(): Gflush(): UsuarioVO[1.*] GfindByld(id:Integer): Optional --> Gexist8yld(id:Integer): boolean deleteAllInBatch () getOne (id: Integer): UsuarioVO getime (a: imeger): UsuanovU count (): long defletelyId (id: Integer) getByld (id: Integer): UsuarioVO getByld (id: Integer): UsuarioVO delete (entity : UsuarioVO) save (entity: \$1: \$ --saveAll (entities : Iterable) : SI1..*1=... saveAllAndFlush (entities : Iterable) : S[1..*] -...> findAllByld (ids: Iterable): UsuarioVO[1.*] deletelnBatch (entities : Iterable) deletentiblatch (entities: Rerable) findAll (exemple: Example, pageable: Pageable): Page «...» deleteAllinBatch (entities: Resable) count (example: Example): long «...» deleteAllBytdinBatch (ids: Iterable) exists (example : Example) : boolean «...» findBy (example : Example , queryFunction : Function) : R «...» deleteAllByld (ids : Iterable) deleteAll (entities : Sterable) | GridAll (example : Example) : S[1,*]+...* | findAll (example : Example , sort : Sort) : S[1,*]+...*

ServicioUsuarioImpl ur: UsuarioRepository G findByCorrea (correa : String) : Optional --flush() e...» findAll(): UsuarioVO[1..*]+...+ og findByld (id : Integer) : Optional «...»

og existsByld (id : Integer) : boolean «...»

og deleteAllinBatch () «...»

og getOne (id : Integer) : UsuarioVO «...» count(): long +...+ deleteByld (id : Integer) getByld (id: Integer): UsuarioVO getReferenceByld (id : Integer) : UsuarioVO +...+ getiriderenceby(d (d:linteger): Usuan gelete(entity: UsuarioVO) «...» g save(entity: S): S «...» g saveAll(entities: Iterable]: S[1..*] «...» findOne (example : Example) : Optional *... >
findAll (sort : Sort) : UsuarioVO [1.."] <... > findAll (pageable : Pageable) : Page saveAndFlush (entity: S): S +...+ \$ asveAllAndFlush (entity): 5) : 5 «...»
\$ soveAllAndFlush (entities : iterable) : 5 | 1..."| «...»
\$ findAllById (ids: Iterable) : UsuarioVO [1..."| «...»
\$ deleteinBatch (entities : Iterable) «...»
\$ findAll (example : Example , pageable : Pageable) : Page «...» count (example : Example) : long ---deleteAlfBvidInBatch (ids : herable) ----G detectAlBydnBatch (ds: Brabble) «...»

Gestife (sample | Looplets «...»

Gestife (sample | Looplets «...»

Gestife (sample | Looplets »...»

Gestife (sample | Looplets »...»

Gestife (sample : Example) «...»

Gestife (sample : Example) »...»

Gestife (sample : Example) : S[1...] «...» findAll (example : Example . sort : Sort) : S[1_1] ---



count (example : Example) : long «...»
delete Alfily (div Batch (ids : Iterable) «...» findAll (example : Example) : S[1.*] «... findAll (example: Example, sort: Sort.): S[1_*1=_= () flush () =...»
() findAll () [1...") =...»
() findByld (id: Integer): Optional =...»
() existsByld (id: Integer): boolean =...»
() deleteAllinBatch () =...» ServicioFiercicio (Q flush() G_findAll(): EjercicioVO[1..*]
G_findByld(id:Integer): Optional «...» getOne (id: Integer) sur count(): long =_= deleteByld (id : Integer) --count(): long deleteByld (id : Integer) getByld (id : Integer) : EjercicioVO getReferenceByld (id : Integer) : EjercicioVO G_getFeferenceByld (id: Integer): EjercicioW
G_geteleAll ()
G_geteleComplete (): Optional =...=
G_geteleComplete (): SizercicioVO)
G_get save (entity: 5): 5 +... saveAll (entities : Iterable) : \$11_*1 a...s findOne (example : Example) : Optional +... findAll (sort : Sort) [1..*] +...+ indOne (example : Example) : Optional «...»
findAll (sort : Sort) : EjercicioVO [1_*] Comman(sont sont) : bgrecciovO([1-7]

Goffman(sont sont) : bgrecciovO([1-7]

Goffman(sont) : bgrecciovO([1 deleteinBatch (entities : herable) <...> findAll (example : Example , pageable : Pageable) : Page +...+ deleteinBatch (entities : Iterable) findAll (example : Example , pageable : Pageable) : Page +...+ deleteAllinBatch (entities : Iterable) count (example : Example) : Inno a.... ्ु oetecansyndnisators (us: iteratore) ु exists (example : Example): boolean काल ु findBy (example: Example, queryFunction : Function) : R काल ु deleteAlById (ids : Iterable) deleteAllByld (ids: Iterable) a...» deleteAll (entities : Iterable) findAll (example : Example) : S[1_"] <-->
findAll (example : Example , sort : Sort) : S[1..*] <--deleteAll (entities : Iterable)

ServicioEjercicioImpl

er: EjercicioRepository

findAll(): EjercicieVO[1_*] ...> findByld (id: Integer): Optional «...»

cipcistsByld (id: Integer): boolean «..»

deleteAllnBatch() «...»

getOne (id: Integer): EjercicioVO «...»

Count(): long ---

getByld (id : Integer) : EjercicioVO «...»

G_getRerenceByld (id : Integer): EjercicioV
G_deletcAll () =..»
G_deletc_All () =..»
G_deletc_entby: EjercicioVO) =...»
G_save(entby: 5): S=...»
G_save(entby: 5): S=...»

findOne (example : Example) : Optional + ...

(3_fmcMall (post 5 or) 1; feer convO(1.1*) == (5_fmcMall (post 5 or) 2; feer convO(1.1*) == (5_fmcMall (post 5 or) 2; feer convO(1.1*) == (5_fmcMall (post 6 or) 2; feer convO(1.1*) == (5_fmcMall (convo)(1.1*) == (5_fmcMall (convo)(1

findAll (sort : Sort) : EjercicioVO[1..*] +...+

deleteAllinBatch (entities : Iterable) +...+

(flush () = ...=

```
pr: PreguntaRepository
                        findByld (id : Integer ) : Optional +...+
                           existsByld (id : Integer ) : boolean +...+
                           _deleteAllinBatch() «...»
                     C_deleteAllnBatch () = ~...
C_getOne (id : Integer ): PreguntaVO =...x
C_gcount (): Inng =...
C_deleteByld (id : Integer ) = ...x
C_getByld (id : Integer ): PreguntaVO =...
C_getByld (id : Integer ): PreguntaVO =...
C_getByld (id : Integer ): PreguntaVO =...
                           deleteAll() +...+
delete(entity:PreguntaVO) -...-
                              FindByEjercicio (ejercicio : EjercicioVO ) : Optional «...»
                  saveAndFlush (entity : S ) : S ---
                              seveAltEndFlush (entities : Berable ) : $11.*Lu...
                           findAllByld (ids : Iterable ) : PrepuntaVQ [1_*] s_
                              count (example : Example ) : long «...»
                              deleteAlByldinBatch (ids: Iterable ) ....
                                    exists (example : Example ) : boolean »...
            G_Beist (example: Example ) boolean «...»

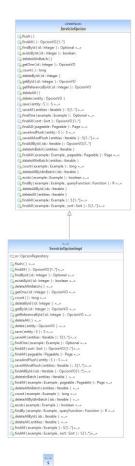
G_Briddy (example: Example, queryfruction ; Function ) : R «...»

G_Briddy (example: Example, queryfruction ; Function ) : R «...»

G_Briddy (example: Example ) »...»

G_Briddy (example: Example ) : S[1...*] «...»

G_Briddy (example: Example : Sort Sort ) : S[1...*] «...»
iflush()
findAll(): PreguntaVO[1.*]
findByld(id: Integer): Optional
existsByld(id: Integer): boolean
         deleteAllinBatch ( )
   cg_elekteAllintsitch()
Gg_egfOne(dicInteger): PreguntaVO
Gg_count(): long
Gg_elekteByld(id:Integer): PreguntaVO
Gg_etByld(dicInteger): PreguntaVO
Gg_etByld(dicInteger): PreguntaVO
            deleteAll()
            delete (entity : PreguntaVO )
      @ delete (enthy: PreguntaVO )
@ findlify processing (eprocise) (ep
            saveAndFlush (entity: 5 ): 5 +...
                  seveAllAndFlush (entities : Iterable 1: SI1.-*1-...»
            seventionathistri (entities : iterable ) : $[1...]*...
findAllByld (ids : Iterable ) : PreguntaVO [1...]
deleteinBatch (entities : Iterable )
findAll (example : Example , pageable : Pageabl
deleteAllInSatch (entities : Iterable )
                     count ( example : Example ) : long «...»
               deleteAllBvidInBatch (ids : Iterable )
               exists (example : Example ) : boolean +...
                  findBy (example : Example , queryFunction : Function ) : R -> deleteAllById (ids : Iterable )
   Conclusion of the state of the
```





findByUsuario (user: UsuarioVO): Optional a... og hndbyGsvano (user: UsuanovO): Optional «...»

og sove (entity: 5): 5 «...»

og taveAll (entities: Herable): S[1...*] «...»

og findOne (example: Example): Optional «...»

og findAll (sort: Sort): UsuarioEjercicioVO [1..*] «...»

findAll (pageable : Pageable) : Page

deleteAllByldinBatch (ids: Iterable) =...a

exists (example : Example) : boolean sure

saveAllAndFlush (entities : Iterable) : S[1..*] =...

G_ saveAllAndFluch (enthies: Iterable): 5 (1-1" n...»

GrinfallByld (5s: Iterable): 1 (2mois)Epriciol(0 (1...") n...»

GeleteinBatch (enthies: Iterable) n...»

Grinfald (example: Example, pagable: Pageible): Page n...»

GeleteinBullSelot (enthies: Iterable) n...»

Gount (example: Example): long n...»

Cg. exists (earngle: Eximple, people): 1. boolean*

G. findBy (earngle, Example, people function): R e...*

G. deleteAllByld (ids: Iterable) = ...*

G. deleteAll (entities: Breakle) = ...*

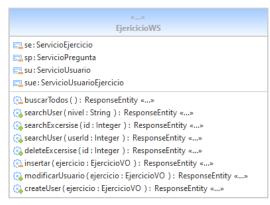
G. findBat (oanngle Example): \$151...*| e...*

G. findBat (oanngle Example): \$151...*| e...*

saveAndFlush (entity: S 1: S sue



WS



UsuarioWS

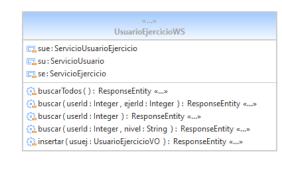
su: ServicioUsuario

searchAll(): ResponseEntity «...»

😪 searchUser (correo : String) : ResponseEntity «...»

🔐 createUser (usuario : UsuarioVO) : ResponseEntity «...»

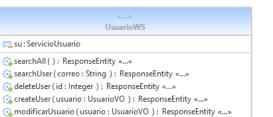
@ deleteUser (id : Integer) : ResponseEntity «...»





CorsConfig

😘 addCorsMappings (registry : CorsRegistry) «...»

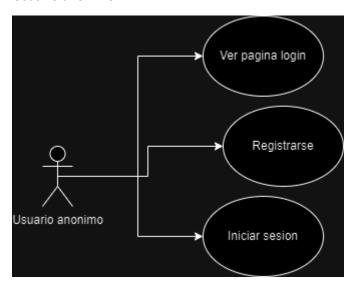


PreguntasWS 🔁 sp : ServicioPregunta se: ServicioEjercicio 😘 searchUser (idEjercicio : int) : ResponseEntity «...» 🎧 searchQuestion (idPregunta : int) : ResponseEntity «...» 😘 deleteQuestion (id : Integer) : ResponseEntity «...» modificarPregunta (pregunta: PreguntaVO): ResponseEntity «...» 🔐 createPreg (pregunta : PreguntaVO , idEjercicio : Integer) : ResponseEntity «...»

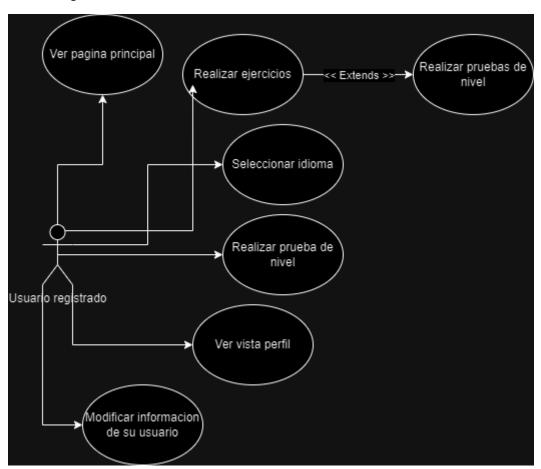
OpcionWS so: ServicioOpcion sp: ServicioPregunta 🕞 deleteQuestion (id : Integer) : ResponseEntity «...» nodificarPregunta (opcion: OpcionVO): ResponseEntity «...» CreatePreg (opcion: OpcionVO, idPregunta: Integer): ResponseEntity «...»

Diagrama casos de uso

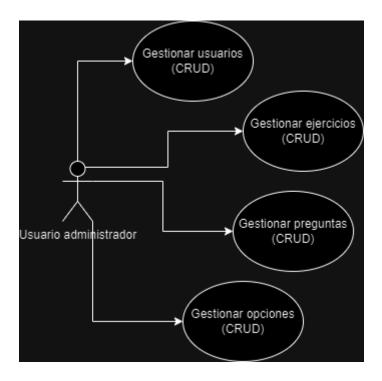
Usuario anónimo



Usuario registrado



Usuario administrador



Tecnologías aplicadas

Back-end:

- Java 17: lenguaje de programación.
- MySql: Gestor de base de datos
- Maven: Gestor de dependencias
- Spring boot: Framework para el desarrollo de aplicaciones web en Java
- Lombok: Librería que permite automatizar la generación de código repetitivo

Front-end:

- HTML5: Lenguaje de marcado para la creación de páginas web
- CSS3: Lenguaje de marcado para dar estilo a las páginas web
- **JavaScript:** Lenguaje de programación en entorno cliente para dar funcionalidad a páginas web
- FontAwesome: Librería para iconos utilizados en la aplicación.
- Cloudflare: Librería con servicios de seguridad (utilizada para todo el proceso de encriptación de información de la aplicación)

Bibliografía

ChatGPT: https://chatgpt.com/

StackOverflow: https://es.stackoverflow.com/