

Grau en Enginyeria Informàtica de Gestió i Sistemes d'Informació

APLICACIÓ DE CREACIÓ I SEGUIMENT DE RUTINES D'EXERCICI

Memòria

PAU MENINO SUREDA
TUTORA: CATALINA JUAN NADAL

CURS 2022-2023

Agraïments

A totes aquelles persones que han cregut en mi.

Abstract

Currently, the demand for mobile applications in the exercise and health sectors is growing. More and more people are concerned about their physical well-being and seek tools to help them improve their performance and achieve their training goals efficiently. Considering the widespread use of mobile technology, the objective of this project is to provide an innovative and useful solution for people interested in improving their physical performance and taking care of their health through the development of a hybrid mobile application that allows the creation of customized exercise routines and tracking of the progress made in each exercise.

Resum

Actualment, la demanda d'aplicacions mòbils en el sector de l'exercici i la salut es troba en creixement. Cada vegada són més les persones que es preocupen pel seu benestar físic i busquen eines que els ajudin a millorar el seu rendiment i assolir els objectius d'entrenament de manera eficient. Tenint en compte el gran ús de la tecnologia mòbil, l'objectiu d'aquest projecte és proporcionar una solució innovadora i útil per a les persones interessades en la millora del rendiment físic i cuidar la salut, tot això mitjançant el desenvolupament d'una aplicació mòbil híbrida que permeti la creació de rutines d'exercici personalitzades i el seguiment de la progressió realitzada en cada exercici.

Resumen

Actualmente, la demanda de aplicaciones móviles en el sector del ejercicio y la salud se encuentra en crecimiento. Cada vez son más las personas que se preocupan por su bienestar físico y buscan herramientas que les ayuden a mejorar su rendimiento y lograr los objetivos de entrenamiento de manera eficiente. Teniendo en cuenta el gran uso de la tecnología móvil, el objetivo de este proyecto es proporcionar una solución innovadora y útil para las personas interesadas en la mejora del rendimiento físico y cuidar la salud, todo ello mediante el desarrollo de una aplicación móvil híbrida que permita la creación de rutinas de ejercicio personalizadas y el seguimiento de la progresión realizada en cada ejercicio.

Índex

| | |
|---|-----|
| Índex de figures | III |
| Índex de taules..... | V |
| Glossari de termes | VII |
| 1. Objecte del projecte. | 1 |
| 1.1. Motius i motivacions. | 1 |
| 2. Marc teòric i anàlisi de referents..... | 3 |
| 2.1. Antecedents..... | 3 |
| 2.1.1. Jefit. | 3 |
| 2.1.2. Strong. | 4 |
| 2.1.3. FitNotes. | 5 |
| 2.2. Necessitats d'informació. Teoria de l'exercici. | 7 |
| 2.2.1. Repeticions i sèries. | 8 |
| 2.2.2. Fallada muscular..... | 9 |
| 2.2.3. Sobrecàrrega progressiva..... | 10 |
| 2.3. Necessitats d'informació. Tecnologies. | 12 |
| 2.3.1. Spring Boot..... | 12 |
| 2.3.2. Node.js..... | 13 |
| 2.3.3. Ionic React..... | 14 |
| 2.3.4. Visual Studio Code..... | 15 |
| 2.3.5. IntelliJ IDEA. | 17 |
| 3. Objectius i abast. | 19 |

| | |
|---|----|
| 4. Metodologia..... | 21 |
| 5. Desenvolupament..... | 29 |
| 5.1. Definició de requeriments funcionals i tecnològics. | 29 |
| 5.2. Base de dades..... | 30 |
| 5.2.1. Disseny..... | 30 |
| 5.2.2. Implementació..... | 33 |
| 5.3. Aplicació mòbil. | 37 |
| 5.3.1. Disseny..... | 37 |
| 5.3.2. Implementació..... | 42 |
| 6. Possibles ampliacions. | 55 |
| 7. Anàlisi de resultats i conclusions..... | 57 |
| 7.1. Resultats de la metodologia..... | 57 |
| 7.2. Resultats de les tecnologies. | 58 |
| 7.3. Conclusions. | 59 |
| 8. Bibliografia. | 61 |

Índex de figures

| | |
|--|----|
| Fig. 2.1.1.2. Exemple d'una rutina d'entrenament mitjançant l'aplicació Jefit | 4 |
| Fig. 2.1.2.1. Exemple del seguiment de la progressió d'un exercici mitjançant l'aplicació Strong | 5 |
| Fig. 2.1.3.1. Interfícies de seguiment de la progressió dels exercicis d'una rutina mitjançant l'aplicació FitNotes | 6 |
| Fig. 2.2.2.1. Esquema de la progressió del nivell d'esforç realitzar per realitzar una repetició rere altre fins a arribar a la fallada muscular | 10 |
| Fig. 2.2.3.1. Beneficis de la sobrecàrrega progressiva | 11 |
| Fig. 2.3.2.1. Esquema de les aplicacions de Node.js..... | 14 |
| Fig. 2.3.4.1. Enquesta realitzada per la pàgina Web StackOverflow sobre els entorns més utilitzats per desenvolupadors | 16 |
| Fig. 2.3.5.1. Estructura de la metodologia de desenvolupament àgil..... | 22 |
| Fig. 5.2.1.1. Model lògic de la base de dades..... | 32 |
| Fig. 5.2.1.2. Model relacional de la base de dades | 33 |
| Fig. 5.2.2.1. Estructura per capes del funcionament de JPA | 34 |
| Fig. 5.2.2.2. Fragment de la classe <i>Exercise</i> amb anotacions de JPA..... | 35 |
| Fig. 5.2.2.3. Anotacions de JPA per establir la relació entre les entitats <i>Exercise</i> i <i>Set</i> | 36 |
| Fig. 5.3.1.1. Esquema de paquets del Back End..... | 38 |
| Fig. 5.3.1.2. Disseny de la interfície d'autenticació d'usuari | 39 |
| Fig. 5.3.1.3. Disseny de la interfície d'entrenaments | 39 |
| Fig. 5.3.1.4. Disseny de la interfície de dies d'entrenament | 40 |

| | |
|---|----|
| Fig. 5.3.1.5. Disseny de la interfície d'exercicis d'un dia..... | 41 |
| Fig. 5.3.1.6. Disseny de la interfície de sèries d'un exercici..... | 41 |
| Fig. 5.3.2.1. Codi de la funció de recopilació i emmagatzematge d'exercicis de l'API externa..... | 43 |
| Fig. 5.3.2.2. Codi de les funcions d'inici de sessió i de registre d'usuari..... | 45 |
| Fig. 5.3.2.3. Codi de la classe <i>WebConfig</i> | 46 |
| Fig. 5.3.2.4. Interfície d'inici de sessió..... | 47 |
| Fig. 5.3.2.5. Interfície de registre d'usuari..... | 47 |
| Fig. 5.3.2.6. Interfície de llista de rutines d'un usuari..... | 48 |
| Fig. 5.3.2.7. Interfície de creació d'una nova rutina..... | 49 |
| Fig. 5.3.2.8. Interfície de llista de dies d'una rutina..... | 49 |
| Fig. 5.3.2.9. Interfície de creació d'un nou dia..... | 50 |
| Fig. 5.3.2.10. Interfície de llista d'exercicis d'un dia..... | 51 |
| Fig. 5.3.2.11. Interfície de creació d'un nou exercici..... | 52 |
| Fig. 5.3.2.12. Interfície de llista de sèries d'un exercici..... | 53 |
| Fig. 5.3.2.13. Interfície de creació d'una nova sèrie..... | 54 |

Índex de taules

| | |
|---|----|
| Taula 2.1.3.1. Taula comparativa de les característiques principals ofertes per les aplicacions descrites | 7 |
| Taula 2.3.5.1. Taula comparativa entre la metodologia de desenvolupament àgil i el mètode en cascada..... | 23 |

Glossari de termes

| | |
|---------|--|
| TFG | Treball de Fi de Grau |
| PHP | Llenguatge de programació Preprocessador d'Hipertext |
| C | Llenguatge de programació C |
| C# | Llenguatge de programació C# |
| JS | Llenguatge de programació JavaScript |
| HTML | Llenguatge de Marques d'Hipertext |
| CSS | Llenguatge de Fulls d'Estil en Cascada |
| IDE | Entorn de Desenvolupament Integrat |
| SDK | Conjunt de Desenvolupament de Programari |
| OS | Sistema Operatiu |
| GNU | GNU's Not Unix |
| API | Interfície de Programació d'Aplicacions |
| VS Code | Visual Studio Code |
| GIF | Format d'Intercanvi de Gràfics |
| JPA | API de Persistència de Java |
| SQL | Llenguatge de Consulta Estructurat |
| JDBC | Connectivitat a Bases de Dades de Java |
| HTTP | Protocol de Transferència d'Hipertext |
| CORS | Compartició de Recursos Entre Orígens |

1. Objecte del projecte.

L'objecte o finalitat d'aquest treball de fi de grau consisteix en el desenvolupament d'una aplicació mòbil híbrida que permeti als usuaris registrar o crear les seves rutines d'exercici per tal d'obtenir un seguiment de la progressió dels diversos exercicis a realitzar, tenint en compte les característiques específiques de cada exercici.

1.1. Motius i motivacions.

El desenvolupament d'aplicacions mòbils és un camp en constant creixement i evolució, i actualment hi ha una gran demanda d'aquestes aplicacions a la indústria del fitness i el benestar. Cada cop són més les persones que es preocupen per la seva forma física i busquen eines que els ajudin a millorar el seu rendiment i assolir els seus objectius d'entrenament de manera eficaç.

Tenint en compte aquesta realitat, i tenint en compte l'ús intensiu de les tecnologies mòbils, l'objectiu principal d'aquest TFG és oferir una solució innovadora i útil per a aquells interessats en millorar el rendiment físic i el benestar mitjançant el desenvolupament d'aplicacions mòbils híbrides. L'aplicació permet crear rutines d'entrenament personalitzades i controlar el progrés de cada entrenament.

L'aplicació s'esforça per satisfer les necessitats i expectatives dels usuaris, proporcionant una experiència d'usuari intuïtiva i senzilla que permet als usuaris aconseguir resultats en el menor temps possible. De nou, l'objectiu és crear una eina que pugui ser utilitzada per persones de diferents edats, nivells de condició física i objectius d'entrenament, amb un disseny adaptable per a la facilitat d'ús per a tots els usuaris.

El marc de referència per registrar aquest TFG es basa en l'ús de tecnologies mòbils per millorar la salut i el benestar físic. Amb l'augment de la presència de la tecnologia mòbil en la vida quotidiana de les persones i les creixents preocupacions per la salut i el benestar, aquest camp ha generat un gran interès i temàtiques. A més, s'estan desenvolupant noves tecnologies i aplicacions per a un entrenament i un seguiment de la forma més eficient i personalitzat.

En definitiva, la realització d'aquest TFG té una gran transcendència i potencial per a la millora de la salut física de les persones, proporcionant una eina innovadora i útil per a la creació de programes d'exercici personalitzats i el seguiment del progrés de cada exercici.

2. Marc teòric i anàlisi de referents.

A continuació, es procedeix a contextualitzar el tema en el qual s'emmarca el present TFG. El procés es duu a terme mitjançant una revisió bibliogràfica exhaustiva sobre el tema de recerca, per tal d'establir les bases teòriques i les investigacions prèvies relacionades amb el tema del TFG. Amb aquestes accions tenen com a objectiu la identificació de la problemàtica o el desafiament que s'aborda durant el desenvolupament del present projecte, l'anàlisi de l'estat de l'art en la matèria, incloent-hi les investigacions prèvies i els avenços tecnològics rellevants, establir els objectius de la investigació i proporcionar un marc conceptual clar i una base sòlida per al desenvolupament de la feina.

2.1. Antecedents.

Actualment, degut el gran ús d'internet per part de tot tipus d'usuaris arreu del món, no resulta una complicació descobrir eines digitals o aplicacions que ofereixin allò que cobreixi les necessitats específiques de cadascun. De la mateixa manera, en el cas de les aplicacions orientades a l'esport, es disposa també d'una gran quantitat i varietat d'aplicacions.

Dins d'aquest àmbit, existeixen moltes aplicacions que permeten als seus usuaris rebre un seguiment dels seus entrenaments i la progressió obtinguda en aquests. A continuació, es procedeix a mencionar i descriure algunes de les aplicacions més usades.

2.1.1. Jefit.

Jefit és una aplicació multiplataforma disponible als dispositius Android i Apple. Mitjançant una aplicació web, Jefit també es pot utilitzar mitjançant l'ordinador. És una de les aplicacions més reconegudes dins de l'àmbit de la gestió d'entrenaments.

L'aplicació permet crear plans d'entrenament de forma fàcil, a més de permetre també observar rutines d'entrenament destacades, les quals poden ser adoptades pels usuaris i podran observar les sèries i repeticions corresponents a cada exercici. A més, l'aplicació permet la creació de rutines pròpies mitjançant l'addició dels exercicis desitjats, junt amb el nombre de sèries i repeticions escollit.

En últim lloc, cal destacar que l'aplicació compta amb una opció de versió Prèmium que, mitjançant un pagament mensual, s'obté accés a l'aplicació sense anuncis ni cap mena de publicitat, a més de rebre rutines més desenvolupades i detallades i informes sobre els progressos de l'entrenament més acurats en comparació amb la versió gratuïta.

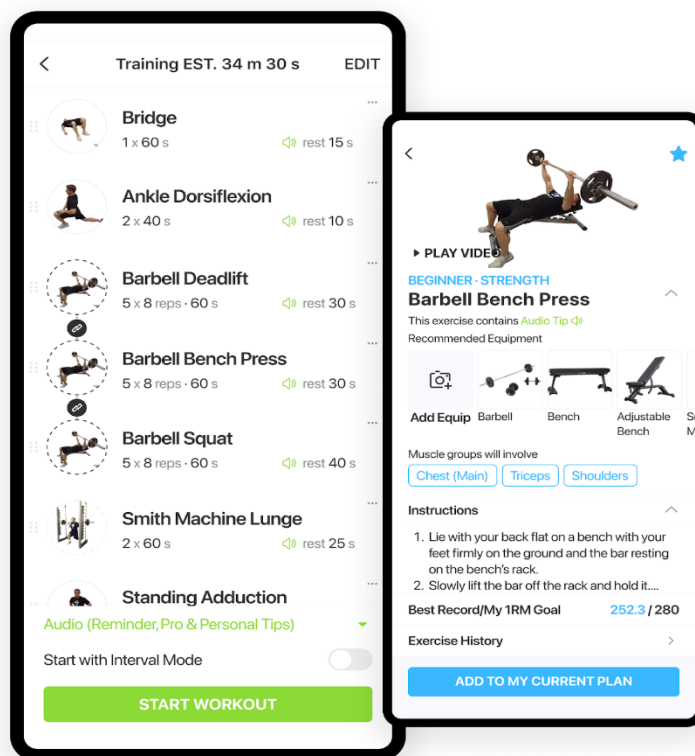


Fig. 2.1.1.1. Exemple d'una rutina d'entrenament mitjançant l'aplicació Jefit. Font:

<https://www.jefit.com/>

2.1.2. Strong.

Strong és una aplicació mòbil disponible pels dispositius Android i iOS. Es troba classificada com una de les aplicacions esportives més destacades de tot el mercat d'aplicacions mòbils dins d'aquest àmbit.

L'aplicació ofereix una gran diversitat de funcionalitats per tal d'adaptar-se a les necessitats de l'usuari, incloent-hi la capacitat de gravar els entrenaments de forma fàcil i intuïtiva, poder rebre i utilitzar plans d'entrenament i exercicis registrats de forma predeterminada. A més, Strong permet enregistrar diversos paràmetres a destacar dins de

l'àmbit de l'entrenament, com la mateixa rutina de l'usuari i les seves mètriques de pes i cos.

De la mateixa manera que amb l'aplicació Jefit descrita prèviament, aquesta també atorga la possibilitat d'efectuar un pagament que permeti als usuaris gaudir de l'aplicació sense publicitat i funcionalitats addicionals.

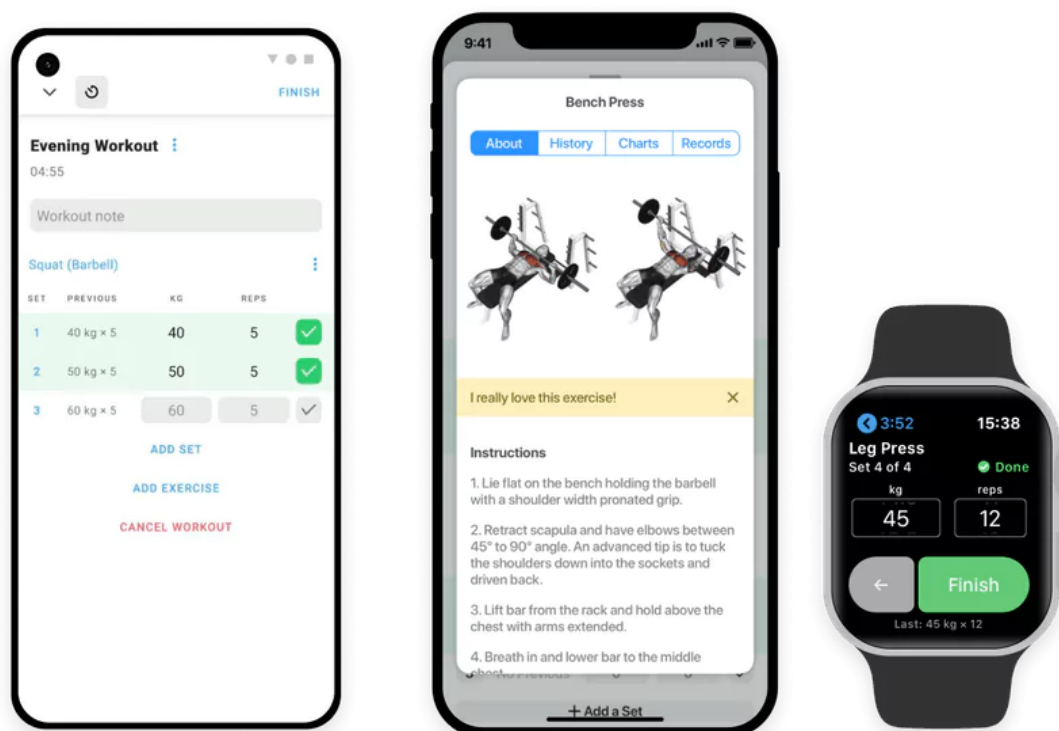


Fig. 2.1.2.1. Exemple del seguiment de la progressió d'un exercici mitjançant l'aplicació Strong. Font: <https://www.strong.app/>

2.1.3. FitNotes.

FitNotes és una aplicació disponible únicament pels dispositius Android i permet dur a terme un control dels entrenaments de forma senzilla i neta.

L'aplicació ofereix als seus usuaris les funcionalitats més imprescindibles a l'hora de realitzar un seguiment dels entrenaments, incloent-hi un cronòmetre que permet controlar la duració dels entrenaments, una llista per afegir els exercicis corresponents a cada dia de la rutina i modificar el nombre de sèries, repeticions i càrrega corresponents i gràfics d'anàlisi de la progressió realitzada a cada exercici.

Finalment, cal destacar que aquesta aplicació no compta amb versió de pagament, fet que implica que qualsevol usuari pot fer servir tot el conjunt de funcionalitats que ofereix l'aplicació tan bon punt l'instal·la al seu dispositiu.

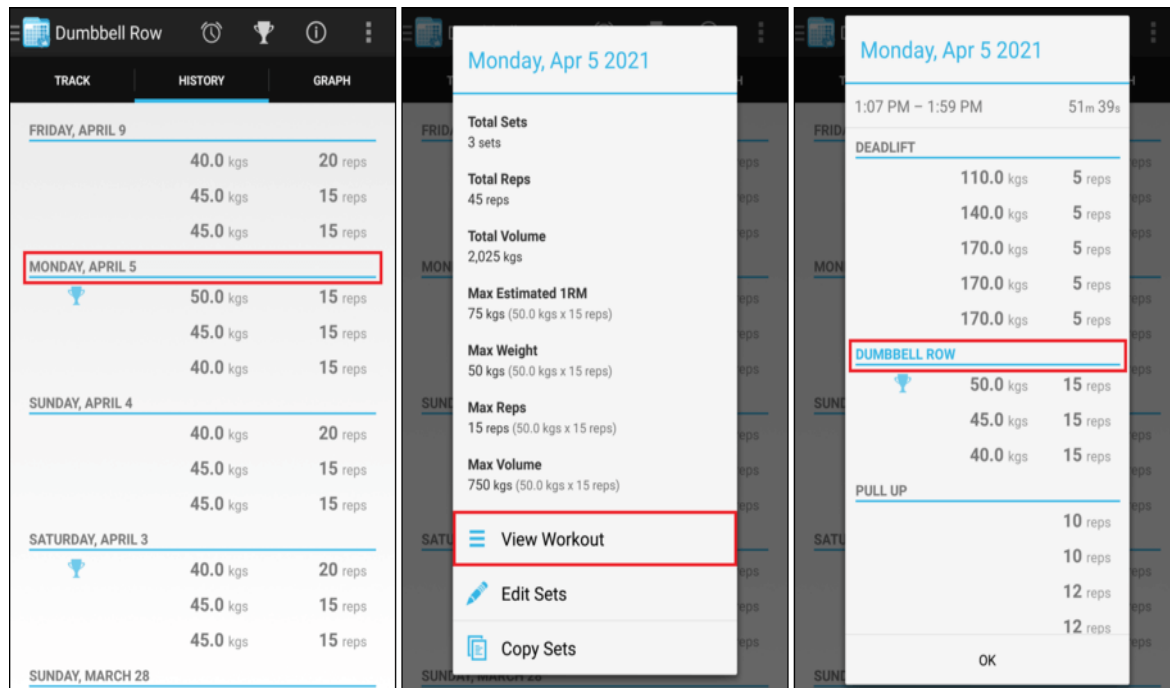


Fig. 2.1.3.1. Interfícies de seguiment de la progressió dels exercicis d'una rutina mitjançant l'aplicació FitNotes. Font: <http://www.fitnotesapp.com/>

Una vegada s'han descrit tres de les aplicacions més utilitzades en l'àmbit esportiu, s'ha elaborat una anàlisi comparativa de les diverses característiques principals que ofereixen:

| | Jefit | Strong | FitNotes |
|--|--------------|---------------|-----------------|
| Rutines d'entrenament predeterminades | ✓ | ✓ | X |
| Seguiment de la progressió dels exercicis | X | ✓ | ✓ |
| Disponibilitat a diverses plataformes | ✓ | ✓ | X |
| Addició d'exercicis no existents de forma predeterminada | X | X | ✓ |

Taula 2.1.3.1. Taula comparativa de les característiques principals ofertes per les aplicacions descrites. Font: Elaboració pròpia.

Tal com es pot observar, la problemàtica principal detectada tracta de la impossibilitat de rebre una rutina d'entrenament personalitzada, és a dir, en funció de certs paràmetres que estableixi l'usuari. A més, s'han identificat també altres limitacions que es presenten en les diverses aplicacions, com no poder rebre rutines d'entrenament predeterminades, no poder realitzar un seguiment de la progressió dels exercicis o bé la no disponibilitat a les diverses plataformes mòbils.

Per aquest motiu, el present TFG planteja el desenvolupament d'una aplicació mòbil que no contingui les limitacions esmentades, per tal d'oferir un producte que permeti diferenciar-se de la resta d'aplicacions existents en el mateix àmbit.

2.2. Necessitats d'informació. Teoria de l'exercici.

En aquest apartat es donen a conèixer a escala teòrica els elements més importants a tractar sobre el projecte dins del camp de l'exercici. El coneixement dels següents conceptes és

una condició necessària per a entendre el funcionament general i propòsit de l'aplicació a desenvolupar durant el projecte.

2.2.1. Repeticions i sèries.

Les repeticions s'entenen com el nombre de vegades seguides que es repeteix un mateix exercici fins a realitzar el descans. El conjunt de repeticions anteriors a cada descans és el que es denomina com a sèrie [1].

Una rutina d'entrenament es constitueix mitjançant una llista d'exercicis, els quals es compleixen mitjançant un conjunt de sèries i repeticions assignades. Per exemple, si un exercici té assignades quatre sèries de quinze repeticions, significa que l'usuari ha de realitzar l'exercici quinze vegades seguides i descansar durant quatre cops seguits. Cal destacar la necessitat de realitzar una pausa de descans entre sèries, de període variable en funció dels objectius de la persona, per tal de poder realitzar la propera sèrie de la forma més bona i amb la menor fatiga possible.

Existeixen molts tipus de sèries que s'enfoquen en diversos paràmetres, tenint en compte l'objectiu de la persona, circumstàncies i nivell, però es recomana que un entrenador personal s'encarregui d'assignar aquests tipus de sèries per tal d'optimitzar els entrenaments de cada persona.

Els tipus de sèries més populars i utilitzats són els següents [1]:

- **Supersèries:** Inclouen la realització de dos exercicis diferents sense descans. Els exercicis han d'estar enfocats a músculs diferents. Després de la sèrie, es fa un descans i es segueix amb la sèrie següent. Aquest tipus de sèries són molt efectives per a persones que no disposin de gaire temps per entrenar, ja que permet treballar un múscul mentre l'altre descansa.
- **Sèries piramidals:** Poden ser de tipus ascendent o descendent. A les sèries ascendents, l'objectiu és reduir el nombre de repeticions i augmentar el pes a utilitzar, en comparació amb la sèrie anterior. Per altra banda, en el cas de les sèries piramidals descendents es tracta d'augmentar les repeticions i reduir el pes en comparació amb la sèrie anterior.

- Sèries dobles i sèries triples: Aquestes sèries tracten d'encadenar diferents exercicis que treballin el mateix grup muscular i sense parar entre ells. S'anomenen sèries dobles aquelles sèries que encadenen dos exercicis i es parla de sèries triples quan s'encadenen tres exercicis.
- Sèries gegants: Les sèries gegants tenen el mateixos objectius que les sèries dobles i sèries triples, només que en aquest cas es combinen quatre o cinc exercicis diferents seguits, amb molt poc descans o sense.
- Sèries ardents: Les sèries ardents consisteixen a realitzar repeticions parcials (el moviment de l'exercici de forma parcial) al final de cada sèrie, de forma que es tradueix en un esforç addicional pel múscul treballat.

2.2.2. Fallada muscular.

La fallada muscular es defineix com la incapacitat momentània del múscul que està sotmès a una càrrega per poder continuar generant la tensió necessària per superar la resistència imposada per la mateixa càrrega, produint-se en la fase del moviment on menys força es pot generar, traduïnt-se en el moment de la repetició en la qual no es pot seguir el moviment [3].

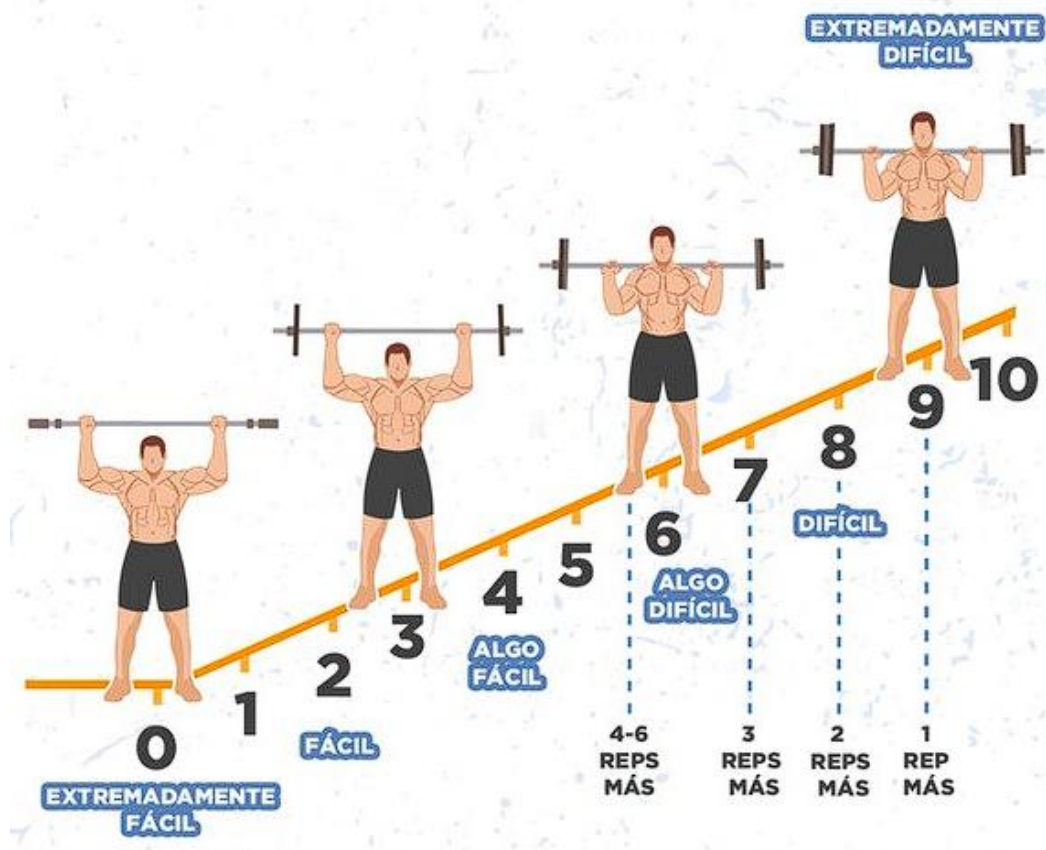


Fig. 2.2.2.1. Esquema de la progressió del nivell d'esforç realitzar per realitzar una repetició rere altre fins a arribar a la fallada muscular. Font:

<https://fullmusculo.com/sobrecarga-progresiva/>

2.2.3. Sobrecàrrega progressiva.

La sobrecàrrega progressiva defineix el concepte de l'augment gradual de l'esforç al qual és sotmès el cos durant l'entrenament físic amb l'objectiu de generar les adaptacions necessàries per poder rebre cada cop més estímuls gràcies als guanys de força o capacitat per aplicar la força de forma més efectiva [5].

L'aplicació d'aquesta tècnica permet continuar evolucionant en cada entrenament i evitar l'estancament. Per tal de dur a terme aquesta estratègia, existeixen diverses variables a aplicar en funció dels objectius i condició física de cada persona.



Fig. 2.2.3.1. Beneficis de la sobrecàrrega progressiva. Font:

<https://fullmusculo.com/sobrecarga-progresiva/>

Les variables de sobrecàrrega progressiva més usades són les següents [4]:

- Repeticions i sèries: Realitzar cada vegada més repeticions amb la mateixa càrrega o pes. També es pot augmentar el nombre de sèries realitzades per exercici.
- Càrrega: Augmentar progressivament el pes, ja que a mesura que es guanya força, resulta més senzill moure una determinada càrrega.
- Freqüència: Augmentar el nombre de sessions d'entrenament o augmentar les vegades que s'entrena cert grup muscular.
- Temps sota tensió: Augmentar el temps de contracció muscular amb l'objectiu de mantenir el múscul treballant durant més temps per cada repetició.
- Descans entre sèries: En funció del propòsit de cada persona es pot modificar el temps de descans entre sèries, reduint-lo de tal forma que es puguin realitzar més

sèries en el mateix temps o bé augmentant-lo per poder realitzar sèries més exigents a nivell de càrrega.

Un cop descrits els conceptes dins del camp de l'exercici necessaris per a la realització del projecte, és imperatiu situar els elements dins del context del propòsit del projecte. Tal com s'ha esmentat, l'objectiu del present TFG consisteix en el desenvolupament d'una aplicació que permeti crear rutines d'exercici i realitzar un seguiment de la progressió en cada exercici.

En termes generals, per tal d'aconseguir obtenir una progressió en els exercicis d'una rutina esportiva i establir una sobrecàrrega progressiva en cadascun d'aquests, cal assegurar que les sèries realitzades siguin efectives. Una sèrie es considera efectiva quan el nombre de repeticions realitzades durant la sèrie s'ha acostat molt al llindar de la fallada muscular. Per exemple, si en una sèrie d'un exercici específic amb l'ús d'una càrrega específica s'arriba a la fallada muscular a les 12 repeticions, es pot considerar que la sèrie ha estat efectiva si les repeticions realitzades es comprenen entre nou i onze repeticions, sense arribar a la repetició que causa la fallada muscular per un petit marge.

2.3. Necessitats d'informació. Tecnologies.

En aquest segon apartat sobre les necessitats d'informació es procedeix a descriure de forma teòrica els elements tecnològics més significatius en relació amb el present TFG. El coneixement dels següents conceptes permet comprendre el procediment de desenvolupament a seguir i les diferents eines emprades per a la realització del projecte.

2.3.1. Spring Boot.

La justificació de la utilització de l'eina Spring Boot es pot trobar al punt 4 del document, corresponent a la metodologia.

Spring Boot és un entorn de desenvolupament d'aplicacions web per a Java que s'utilitza per crear aplicacions web escalables i fàcils de mantenir. Aquesta eina proporciona una abstracció de nivell superior sobre altres tecnologies de la plataforma Java, com ara Spring Framework, per tal que els desenvolupadors puguin centrar-se en el desenvolupament de

les funcionalitats de l'aplicació sense haver de preocupar-se per les configuracions o altres aspectes de baix nivell [6], [7].

Les característiques principals de Spring Boot les següents:

- Configuració automàtica: Spring Boot proporciona una configuració automàtica i una integració de tecnologies, la qual cosa significa que la majoria de les aplicacions poden funcionar sense necessitat de configuracions manuals o complicades.
- Integració amb altres tecnologies: Spring Boot s'integra fàcilment amb altres tecnologies de la plataforma Java, com ara Spring Framework, Hibernate, JPA i Servlets.
- Desplegament simplificat: amb Spring Boot es pot crear un fitxer JAR que s'executi automàticament amb totes les dependències necessàries, fet que simplifica el desplegament de l'aplicació.
- Monitoratge de l'estat de l'aplicació: Spring Boot proporciona un mecanisme senzill per supervisar l'estat de l'aplicació, cosa que ajuda a detectar errors i solucionar-los ràpidament.
- Suport per a diferents tipus de bases de dades: Spring Boot proporciona suport per a diferents tipus de bases de dades, incloent-hi MySQL, PostgreSQL, Oracle i MongoDB.
- Seguretat: Spring Boot inclou funcionalitats de seguretat per protegir les aplicacions d'atacs cibernètics.
- Fàcil de provar: amb Spring Boot es pot crear i executar proves de manera senzilla, cosa que ajuda a garantir la qualitat del codi i a detectar errors.

2.3.2. Node.js.

La justificació de la utilització de l'eina Node.js es pot trobar al punt 4 del document, corresponent a la metodologia. Node.js és un entorn de temps d'execució multiplataforma basat en JavaScript especialment dissenyat pel desenvolupament d'aplicacions escalables. Gràcies a les seves característiques, Node.js s'ha convertit en una de les plataformes més populars utilitzades pel desenvolupament d'aplicacions web i mòbils [8].

Els avantatges principals d'aquesta plataforma són els següents:

- Utilitza un llenguatge senzill basat en JavaScript, de manera que els desenvolupadors de diferents llenguatges el poden usar fàcilment.
- Permet gestionar les sol·licituds simultàniament amb un model d'entrada i sortida basat en esdeveniments. • Facilita la creació d'aplicacions web i mòbil altament competitives, pel fet que la codificació incorporada per l'administrador i usuari són semblants.
- Actualment, es considera la plataforma de programari més emprada, superant els entorns d'execució i els llenguatges de programació com PHP i C.
- Posseeix una gran comunitat virtual en constant creixement que ofereix suport i recolza la plataforma.

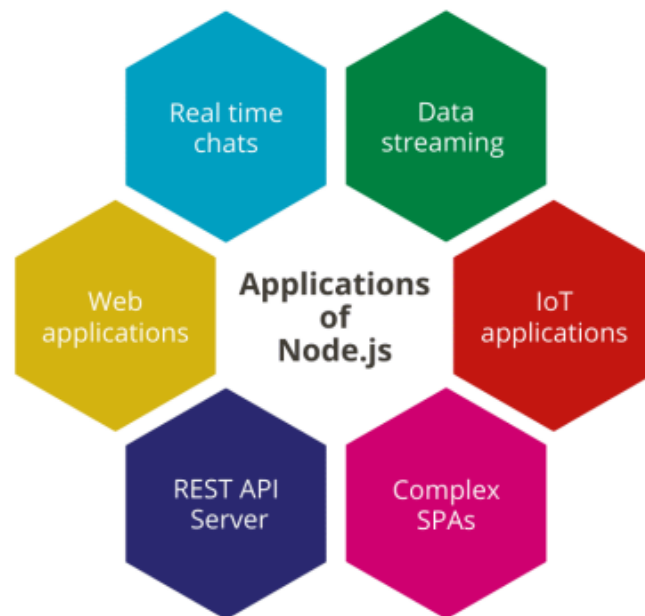


Fig. 2.3.2.1. Esquema de les aplicacions de Node.js. Font:
<https://kinsta.com/knowledgebase/what-is-node-js/>

2.3.3. Ionic React.

La justificació de la utilització de l'eina Ionic React es pot trobar al punt 4 del document, corresponent a la metodologia.

Ionic React és un projecte d'interfície d'usuari i API nativa de codi obert que consta de components d'interfície d'usuari multiplataforma i funcionalitat nativa per crear aplicacions

per iOS, Android, Electron i aplicacions web progressives (aplicacions amb l'objectiu d'oferir als usuaris una experiència igual a les aplicacions natives) mitjançant la tecnologia React i web [9].

Ionic React és la versió oficial de la React de l'entorn Ionic, utilitzat per milions de desenvolupadors web i que alimenta aplicacions de missió crítica per a empreses com T-Mobile, America's Test Kitchen, Schneider, Medical Answering Service, Yara International, IBM i més.

2.3.4. Visual Studio Code.

La justificació de la utilització de l'eina Visual Studio Code es pot trobar al punt 4 del document, corresponent a la metodologia.

VS Code és un editor de codi font desenvolupat per Microsoft. És programari lliure i multiplataforma, de tal manera que es troba disponible per a Windows, GNU/Linux i macOS.

Segons una enquesta realitzada per StackOverflow a més de 80.000 desenvolupadors el maig del 2021, VS Code és l'entorn de desenvolupament més utilitzat, diferenciant-se dels altres entorns de forma destacada [10].

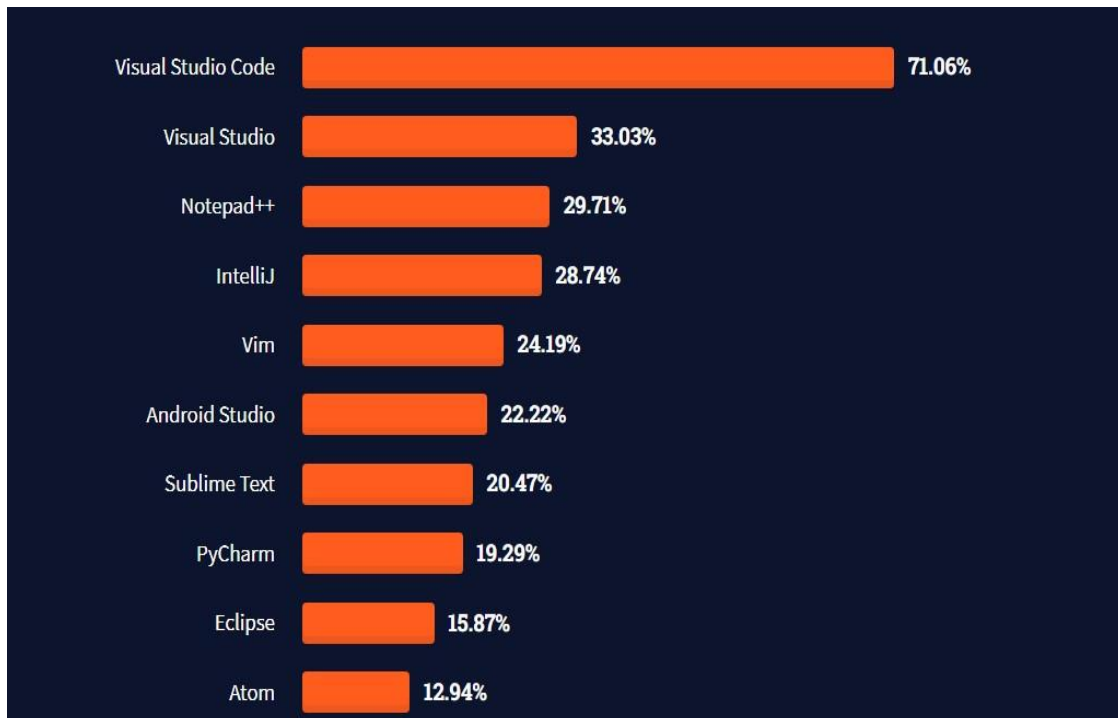


Fig. 2.3.4.1. Enquesta realitzada per la pàgina Web StackOverflow sobre els entorns més utilitzats per desenvolupadors. Font: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

VS Code posseeix una bona integració amb Git, compta amb suport per a depuració de codi i disposa d'un gran nombre d'extensions que ofereix la capacitat de programar i executar codi en qualsevol llenguatge de programació. Les característiques principals de VS Code són les següents:

- Suport integrat per a la codificació amb molts llenguatges, inclosos els que normalment són utilitzats en el desenvolupament multiplataforma, comptant amb funcions d'edició avançades i suport per a llenguatges addicionals mitjançant les possibilitats d'extensió que ofereix l'IDE.
- Depurador integrat dissenyat per a Node.js, encara que també ofereix suport per a depuradors addicionals mitjançant extensibilitat.
- Control de versions basat en Git, que ofereix una experiència integrada de col·laboració. Aquest sistema permet realitzar pujades de codi al repositori i la creació de diverses branques per desenvolupar en paral·lel.

2.3.5. IntelliJ IDEA.

La justificació de la utilització de l'eina IntelliJ IDEA es pot trobar al punt 4 del document, corresponent a la metodologia.

IntelliJ IDEA és un IDE de l'empresa JetBrains per a programació en Java. Ofereix una gran quantitat de característiques que faciliten i acceleren el procés de desenvolupament, incloent-hi:

- Assistència per a codificació.
- Depuració i perfilat de codi.
- Integració amb control de versions i sistemes de construcció.
- Anàlisi i correcció de codi en temps real.
- Plantilles i eines per treballar amb entorns i biblioteques populars.

3. Objectius i abast.

L'abast del projecte engloba el disseny i desenvolupament d'una aplicació mòbil híbrida com a mètode de seguiment de la progressió d'una rutina exercicis pròpia d'un usuari.

És necessari realitzar primerament un estudi i anàlisi de les diverses eines i entorns de desenvolupament, valorant les diverses característiques i avaluant-les de forma que s'escaiguin a les necessitats i requisits del projecte.

Cal construir el disseny de diagrames de flux i de classes com a pas previ a la materialització de l'aplicació mitjançant el seu desenvolupament, per tal d'obtenir una visió clara i concisa dels detalls necessaris per poder discernir i entendre de forma ràpida les tasques a dur a terme, junt amb el procediment a seguir.

És necessari destacar la necessitat de desenvolupar una aplicació de tipus híbrida, per tal de millorar i ampliar el nombre d'usuaris mitjançant un període de desenvolupament curt i menys complex en comparació amb el desenvolupament d'aplicacions úniques per un sistema operatiu. A més, la solució final permet l'enregistrament d'usuaris, de mode que cada usuari pugui accedir amb una garantia de seguretat a la seva pròpia informació.

Un altre punt a destacar sobre el desenvolupament de l'aplicació és que permet a l'usuari registrar la seva pròpia rutina d'exercicis mitjançant una interfície que possibilita la creació d'una llista d'exercicis per dia d'entrenament a manera d'enregistrament de la rutina a la qual es realitza el seguiment de la progressió obtinguda durant cada entrenament (sobrecàrrega progressiva). Per tal d'oferir el seguiment més precís possible, l'aplicació és capaç d'emmagatzemar les característiques de cada exercici, incloent-hi d'aquesta manera el nombre de sèries a realitzar per exercici, les repeticions corresponents a cada sèrie i la càrrega mobilitzada.

Com a últim punt, porta a terme el desenvolupament una funcionalitat que permet a l'usuari rebre una rutina creada per l'aplicació en funció dels paràmetres designats, incloent-hi dies de realització d'exercici a la setmana i nivell de l'entrenament.

A partir de la descripció de l'abast del projecte, es procedeix a descriure el propòsit del treball mitjançant els següents objectius, diferenciant-los en funció de si s'enfoquen al producte o al client:

- Objectius de producte.
 - Desenvolupar una aplicació mòbil híbrida de seguiment de la progressió dels exercicis de rutines dels usuaris.
 - Permetre registrar de forma personalitzada les diverses rutines d'exercici, tenint en compte els possibles exercicis a realitzar, ja siguin usuaris de gimnàs, usuaris d'entrenament a l'aire lliure o bé persones que s'exerciten a casa.
 - Digitalitzar els mètodes de registre dels entrenaments dels usuaris.
 - Introduir a persones que recentment hagin començat a realitzar exercici o que tinguin pensat fer-ho i facilitar el seu procés d'aprenentatge.
 - Oferir rutines d'entrenament als usuaris que vulguin començar a realitzar exercici.
- Objectius de client.
 - Aportar un mode senzill de monitoratge d'exercicis per a les persones que segueixen rutines d'entrenament.
 - Facilitar i agilitzar el procés de registre d'entrenaments dels usuaris.
 - Oferir un mètode de seguiment de la progressió d'exercicis ràpid i intuïtiu.
 - Dissenyar interfícies amb una bona experiència d'usuari.

4. Metodologia.

Per a la realització del projecte s'ha aplicat l'estructura de la metodologia de desenvolupament àgil.

El desenvolupament àgil (també conegut simplement com Agile) és un tipus de metodologia de desenvolupament que anticipa la necessitat de flexibilitat i aplica un nivell de pragmatisme al lliurament del producte. El desenvolupament àgil ofereix un punt de vista diferent en comparació a la visió general que es centra en l'aplicació completa. El punt de vista de la metodologia àgil emfatitza en el lliurament net de peces individuals o parts del programari.

El desenvolupament àgil tracta d'un procés interactiu i incremental. Aquest tipus de desenvolupament es caracteritza per la seva gran capacitat d'adaptació als canvis de requisits, ja que el producte no és especificat completament des de l'inici. A més, aquesta metodologia compta amb diverses entregues de versions i emfatitza en la participació del client.

El concepte de metodologia àgil va sorgir durant la dècada de 1990 com a resposta a les limitacions de la tècnica tradicional de desenvolupament de programari, com el desenvolupament lineal (cascada).

Els primers desenvolupadors de programari àgil varen reunir-se en una conferència anomenada "The Agile Alliance" el febrer de 2001 i van crear els "Principis Àgils" com un marc de treball per tal de poder dur a terme el desenvolupament àgil de programari.

El funcionament de la metodologia àgil es basa en la col·laboració contínua entre els desenvolupadors del projecte, els clients i els usuaris amb l'objectiu de produir un programari final d'alta qualitat mitjançant entregues incrementals de productes funcionals i en la flexibilitat per gestionar canvis dels requisits durant el procés de desenvolupament.

Aquesta metodologia s'ha popularitzat en gran manera dins de la indústria del desenvolupament de programari i s'ha estès a altres àmbits, incloent-hi la gestió de projectes i de productes.

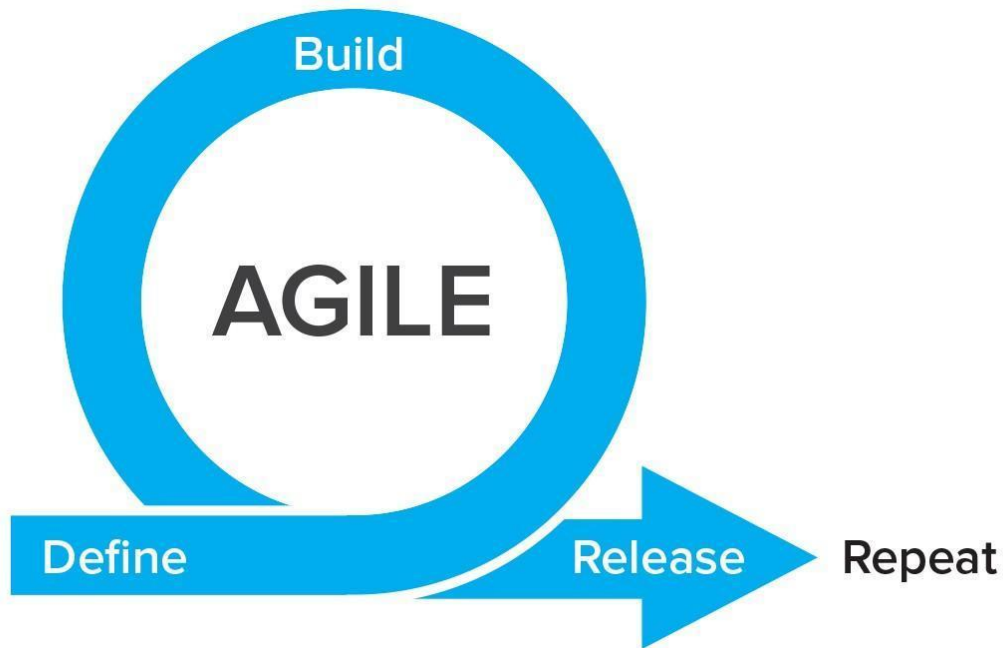


Fig. 2.3.5.1. Estructura de la metodologia de desenvolupament àgil. Font:
<https://www.eqengineered.com/insights/6-considerations-to-ensure-agile-team-value>

Aquesta metodologia, junt amb la metodologia de desenvolupament en cascada, són les tècniques de desenvolupament més utilitzades dins de l'àmbit del desenvolupament de programari. Mitjançant una anàlisi comparativa, s'ha elaborat la següent taula:

| | Àgil | Cascada |
|---------------------------------|--|--|
| Abast del projecte | Els canvis es poden realitzar amb molta antelació amb el temps i dins del pressupost | Funciona bé quan l'abast és conegut des de l'inici i es limiten els canvis |
| Equip | Equips petits o mitjans especialitzats amb alta coordinació | Grans equips, causa un decreixement en la capacitat de coordinació |
| Clients | Permet que els clients estiguin disponibles durant tot el projecte | Només cal que els clients es trobin disponibles a les fites |
| Priorització de funcions | Les característiques es prioritzen i els problemes es resolen segons les prioritats. | No es prioritzen les funcions. |
| Viabilitat | La metodologia àgil funciona millor quan el projecte és factible | No depèn de la viabilitat de projecte |
| Finançament | Augmenta l'eficiència del finançament | Finançament fix mitjançant contractes inicials |

Taula 2.3.5.1. Taula comparativa entre la metodologia de desenvolupament àgil i el mètode en cascada. Font: Elaboració pròpia.

Un cop s'ha realitzat una comparació de les diverses característiques de les dues metodologies al voltant de certes variables, es considera que la metodologia de desenvolupament àgil es mostra com a una millor opció per adoptar pel present projecte, pel fet que s'ajusta més a les necessitats que presenta, constituïdes per la possibilitat de modificar els requisits existents, la necessitat d'establir prioritats i la petita mida de l'equip desenvolupador. Tots aquests fets potencien la necessitat d'ús de la metodologia de desenvolupament àgil.

Per tal d'escollir l'entorn i tecnologia a utilitzar per al desenvolupament del projecte, s'ha elaborat unes anàlisis de les característiques de diverses eines de desenvolupament d'aplicacions, tenint en compte els seus avantatges i desavantatges per tal de contrastar la informació obtinguda amb les necessitats del projecte. A continuació, es mostra una breu descripció de cada eina analitzada, junt amb la llista de pros i contres de cadascuna.

Laravel: Laravel és una eina de desenvolupament mitjançant el llenguatge PHP basada en el model tipus MVC (Model Vista Controlador).

- Avantatges:
 - Usabilitat.
 - Corba d'aprenentatge senzilla.
 - Alta seguretat.
 - Bons sistemes d'autenticació.
 - Comunitat dedicada i codi obert.
 - Plantilles d'interfície.
 - Migració de bases de dades.
 - Arquitectura MVC (Model Vista Controlador).
 - Proves unitàries.
- Desavantatges:
 - Menys funcions en comparació amb altres eines de desenvolupament.
 - Suport limitat.
 - Actualitzacions freqüents que provoquen el mal funcionament de les versions antigues més estables.
 - Funcionament lent.

React Native: React Native és una plataforma de codi obert creada mitjançant JavaScript per Facebook. Permet crear aplicacions mòbils per als sistemes operatius Android i iOS. Poden funcionar en qualsevol dispositiu i utilitzar només una base de codi.

- Avantatges:
 - Usabilitat.
 - Bona experiència d'usuari.
 - Desenvolupament per a qualsevol plataforma i sistema operatiu.
 - Eines de depuració intel·ligents.

- Depuració senzilla.
- Minimalisme.
- Gran comunitat de desenvolupadors.
- Nombroses biblioteques informàtiques de codi obert.
- Compatible amb connectors de tercers.
- Transformació de projectes web a aplicacions mòbils.
- Desavantatges:
 - Baix rendiment en certes plataformes.
 - Ineficiència en cas d'interfícies complexes.
 - Manca de mòduls personalitzats.
 - Problemes d'actualització del propi entorn de desenvolupament.

Ionic: Ionic és un marc multiplataforma de codi obert que permet la creació d'aplicacions mòbils híbrides. Es basa en tecnologies web estandarditzades com HTML, CSS i JavaScript.

- Avantatges:
 - Desenvolupament d'aplicacions híbrides mitjançant una sola base de codi.
 - Aplicacions optimitzades i amb alt rendiment.
 - Estil adaptatiu específic en funció de la plataforma.
 - Àmplia gamma de capacitats d'integració i connectors.
 - Facilitat en la realització de proves.
 - Bona documentació i gran comunitat.
- Desavantatges:
 - Depuració difícil.
 - Aplicacions amb problemes d'execució en temps real.
 - Impossibilitat de modificar el codi sense refrescar l'aplicació sencera.
 - Mides d'emmagatzematge de l'aplicació altes.

Xamarin: Xamarin és una eina que s'utilitza per al desenvolupament d'aplicacions mòbils multiplataforma que utilitza la tecnologia de Microsoft i permet compartir fins a un 90% del codi (C#) a les principals plataformes.

- Avantatges:

- Mateix conjunt de tecnologies per desenvolupar a totes les plataformes.
- Alt rendiment en comparació amb altres eines de desenvolupament híbrid.
- Bona experiència d'usuari.
- Evita errors de compatibilitat de maquinari.
- Tecnologia de codi obert amb un fort suport corporatiu.
- Fàcil manteniment de l'aplicació.
- Desavantatges:
 - Suport retardat per a les últimes actualitzacions de la plataforma.
 - Accés limitat a les biblioteques informàtiques de codi obert.
 - Petita comunitat d'usuaris.
 - Mides d'emmagatzematge de l'aplicació altes.
 - Problemes de compatibilitat amb biblioteques i eines de tercers.
 - Desaparició en un futur pròxim.

Flutter: Flutter és una eina de desenvolupament d'aplicacions en múltiples plataformes i una alternativa sòlida a altres marcs multiplataforma. Flutter funciona amb Dart, un llenguatge de programació orientat a objectes que situa la interfície d'usuari en primer lloc.

- Avantatges:
 - Comunitat gran (i encara en creixement) i suport de Google.
 - Bona optimització i alt rendiment de les aplicacions.
 - Creació d'aplicacions multiplataforma.
 - Canals de plataforma fàcils d'utilitzar que permeten la comunicació directa amb la plataforma nativa.
 - Grans quantitats de biblioteques informàtiques de paquets, API i accessoris disponibles.
 - Creació d'interfícies uniformes que es poden visualitzar exactament igual a totes les plataformes compatibles.
- Desavantatges:
 - Eina reduïda en comparació en l'eina de desenvolupament nadiua.
 - Suport web escàs.
 - Manca de biblioteques de tercers.
 - Poca popularitat de Dart en comparació amb altres llenguatges de programació.

- Mides d'emmagatzematge de l'aplicació altes.

Vue.js: Vue.js es descriu com "un marc progressiu per crear interfícies d'usuari". La seva arquitectura està dissenyada per ser adaptable de manera incremental.

- Avantatges:
 - Fàcil d'aprendre.
 - Versàtil i lleuger.
 - Suport total per la comunitat de codi obert.
 - Arquitectura Model-View-ViewModel (MVVM), millora l'experiència d'usuari de les interfícies i les simplifica, aportant un alt rendiment.
- Desavantatges:
 - Manca d'escalabilitat.
 - Manca de connectors.
 - Poca innovació i manca de nous esdeveniments per part de la comunitat.

Una vegada elaborat l'anàlisi comparativa corresponent per tal de contrastar les diverses característiques que ofereixen les diferents eines, s'ha establert la necessitat d'ús de les eines Ionic i Node.js.

Tal com s'ha comentat, Ionic és un entorn de desenvolupament d'aplicacions mòbils que permet la creació d'aplicacions mòbils híbrides utilitzant tecnologies web com React, HTML, CSS i JS. Aquesta combinació de l'entorn Ionic amb tecnologia React s'anomena Ionic React. Mitjançant això, els desenvolupadors poden crear aplicacions amb una experiència nativa i accés a característiques natives del dispositiu, com ara la càmera i l'emmagatzematge. Per tal d'assegurar el correcte desenvolupament, Ionic React requereix l'entorn Node.js.

Node.js és un entorn d'execució de JavaScript que permet als desenvolupadors crear aplicacions web i servidors web escalables i alta velocitat. Node.js fa ús de JavaScript com a llenguatge de programació i permet als desenvolupadors construir aplicacions amb una arquitectura basada en esdeveniments i una gestió eficient de la concurrència.

En conjunt, Ionic React i Node.js permeten als desenvolupadors crear aplicacions mòbils amb una experiència d'usuari enriquidora i connectar-se a un Back End desenvolupat

mitjançant la tecnologia Spring Boot amb Java. Node.js pot ser usat per construir el servidor i les API necessàries per alimentar l'aplicació mòbil, mentre que Ionic s'encarrega de les interfícies d'usuari i l'experiència de l'aplicació. D'aquesta manera, els desenvolupadors poden crear aplicacions mòbils completes fent ús d'una sola tecnologia (JavaScript) tant al costat del client com al costat del servidor.

Per tal de poder realitzar la programació corresponent i poder utilitzar les tecnologies descrites prèviament, es fa ús dels entorns de programació VS Code i IntelliJ per a la codificació Front End i Back End, respectivament.

5. Desenvolupament.

En aquest apartat es defineix i descriu el format d'estructura de la realització del desenvolupament relacionat a aquest projecte. Tant per la part dels usuaris com per l'aplicació, des del seu disseny fins a la seva implementació.

5.1. Definició de requeriments funcionals i tecnològics.

Per tal d'assolir els objectius i la metodologia a seguir del treball, es procedeix a definir una llista de requeriments concrets funcionals del projecte i requeriments tecnològics, els quals construeixen la base de l'assoliment dels objectius plantejats.

- Permetre als usuaris gestionar la seva sessió.
 - Realitzar l'enregistrament mitjançant operacions d'inici de sessió i tancament de sessió per accedir a la seva informació.
 - Utilitzar un sobrenom com a identificació.
- Permetre als usuaris crear la seva rutina d'exercici.
 - Poder crear rutines d'exercici segons les condicions i gustos (dies i exercicis).
 - Poder consultar contingut demostratiu per realitzar un exercici.
 - Poder modificar la rutina.
 - Poder eliminar les rutines.
- Permetre als usuaris dur a terme un seguiment de la seva progressió als exercicis.
 - Mostrar el nombre de sèries a realitzar corresponent a cada exercici.
 - Indicar la càrrega utilitzada a cada sèrie.
 - Indicar les repeticions realitzades a cada sèrie.

Una vegada definits els requisits funcionals necessaris per al desenvolupament del present TFG, s'ha dut a terme la descripció dels requeriments tecnològics necessaris per poder realitzar les tasques de construcció necessàries. La llista es mostra a continuació:

- IDE per escriure i depurar el codi.
- SDK de la plataforma mòbil per poder accedir a les API i eines necessàries per dur a terme el desenvolupament.

- Simuladors o dispositius mòbils per poder provar i depurar l'aplicació en un entorn real.
- Base de dades per poder emmagatzemar les dades generades i necessàries per al funcionament de l'aplicació.
- Ordinador amb el sistema operatiu Microsoft Windows 10 per poder usar les eines de desenvolupament.

5.2. Base de dades.

El fet de requerir una base de dades es justifica degut la necessitat d'emmagatzemar la informació de la mateixa aplicació, incloent-hi les dades d'usuaris i les rutines d'entrenament creades.

De forma més detallada, l'objectiu principal de la base de dades és l'emmagatzematge de totes les dades d'usuaris, rutines existents, dies corresponents a cada rutina, exercicis assignats a cada dia i les sèries escaients en funció de l'exercici. A més, cal destacar que el Back End carrega una llista d'exercicis a la mateixa base de dades mitjançant una API externa, per tal de tenir un repositori d'exercicis per oferir als usuaris a l'hora d'escollir els exercicis de la seva rutina.

Tenint en compte els aspectes comentats, per tal que la base de dades sigui funcional, aquesta ha de complir els següents requisits:

- Estructura relacional, a causa de la major consistència sobre les dades emmagatzemades en relació amb les estructures no relacionals.
- Sincronització senzilla de les dades del projecte sense la necessitat d'administrar connexions o desenvolupar una lògica de sincronització.
- Capacitat d'integració i compatibilitat amb grans plataformes per a aplicacions mòbils.

5.2.1. Disseny.

El model de base de dades dissenyat s'ha realitzat mitjançant l'eina de modelatge de dades Oracle Data Modeler, que permet als usuaris crear, modificar i mantenir models de dades de manera visual.

Les característiques que ofereix Oracle Data Modeler s'adapten a les necessitats del present projecte, fet que justifica el seu ús. L'eina es caracteritza a través dels següents aspectes:

- Modelatge de dades visual: Permet als usuaris crear models de dades visualment, utilitzant una interfície gràfica que permet la creació i modificació d'objectes de base de dades, com taules, claus, índexs, restriccions i altres elements.
- Suport per a diferents plataformes: L'eina suporta diferents plataformes de bases de dades, incloent-hi Oracle Database, Microsoft SQL Server, IBM DB2, PostgreSQL, MySQL i altres.
- Generació de codi: A partir del model de dades creat, Oracle Data Modeler permet als usuaris generar codi SQL per crear la base de dades i les taules necessàries.
- Integració amb altres eines: Es pot integrar amb altres eines de desenvolupament de bases de dades, com ara Oracle SQL Developer, Eclipse i altres.
- Control de versions: Oracle Data Modeler permet als usuaris gestionar les versions dels models de dades i compartir-los amb altres membres de l'equip de desenvolupament.
- Documentació: L'eina també ofereix funcions per generar documentació del model de dades, com ara descripcions de taules i columnes, relacions i altres elements, que es poden exportar en diferents formats.

A continuació, es mostra el model lògic de la base de dades dissenyat:

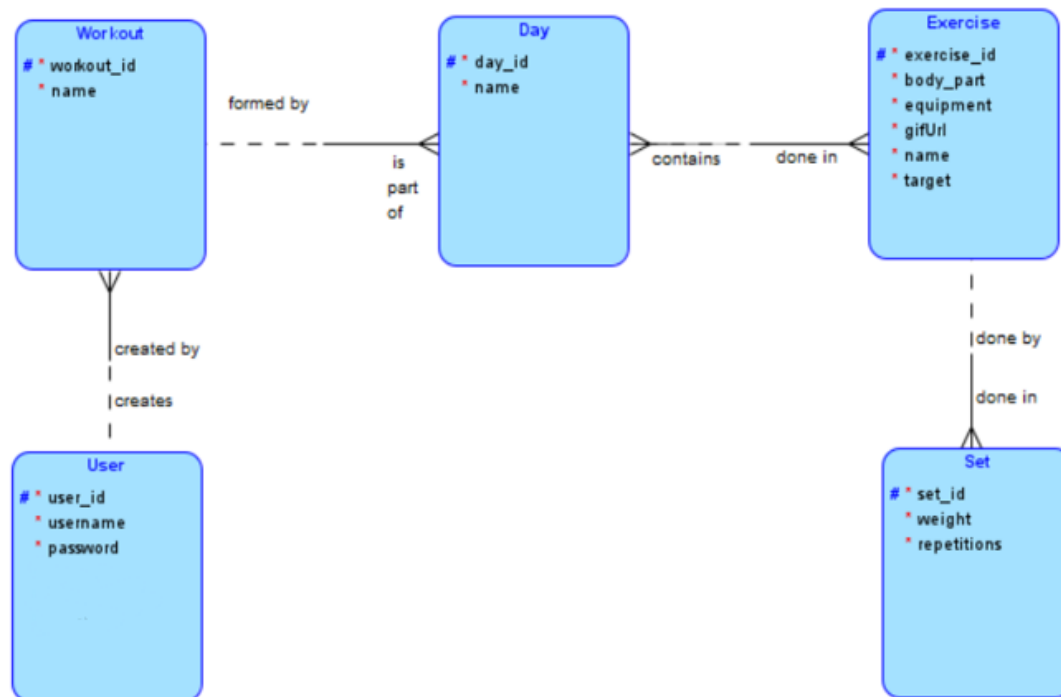


Fig. 5.2.1.1. Model lògic de la base de dades. Font: Elaboració pròpia.

Tal com es pot observar a la figura mostrada, la jerarquia del model es divideix en cinc entitats, anomenades *User*, *Workout*, *Day*, *Exercise* i *Set*.

L'entitat *User* emmagatzema la informació dels usuaris per tal d'assegurar la seva identificació i autenticació. Les entitats *Workout* i *Day* permeten relacionar l'usuari amb els exercicis que esculli, l'entitat del qual (*Exercise*) conté informació sobre els diversos exercicis. La informació de l'entitat esmentada inclou la part del cos involucrada i amb la que es realitza l'exercici, l'equipament necessari per portar-lo a terme, un enllaç de visualització d'un GIF animat de l'execució de l'exercici i el seu propi nom. Finalment, l'entitat *Set* emmagatzema informació sobre les característiques de les sèries dutes a terme als diversos exercicis, per tal de poder visualitzar el nombre de repeticions a realitzar i el pes amb el qual efectuar la sèrie en qüestió.

Cada entitat conté el seu identificador únic, que permet establir relacions entre les diverses entitats i permetre el reconeixement de cadascuna.

A partir del model lògic, s'ha obtingut el següent model de dades relacional:

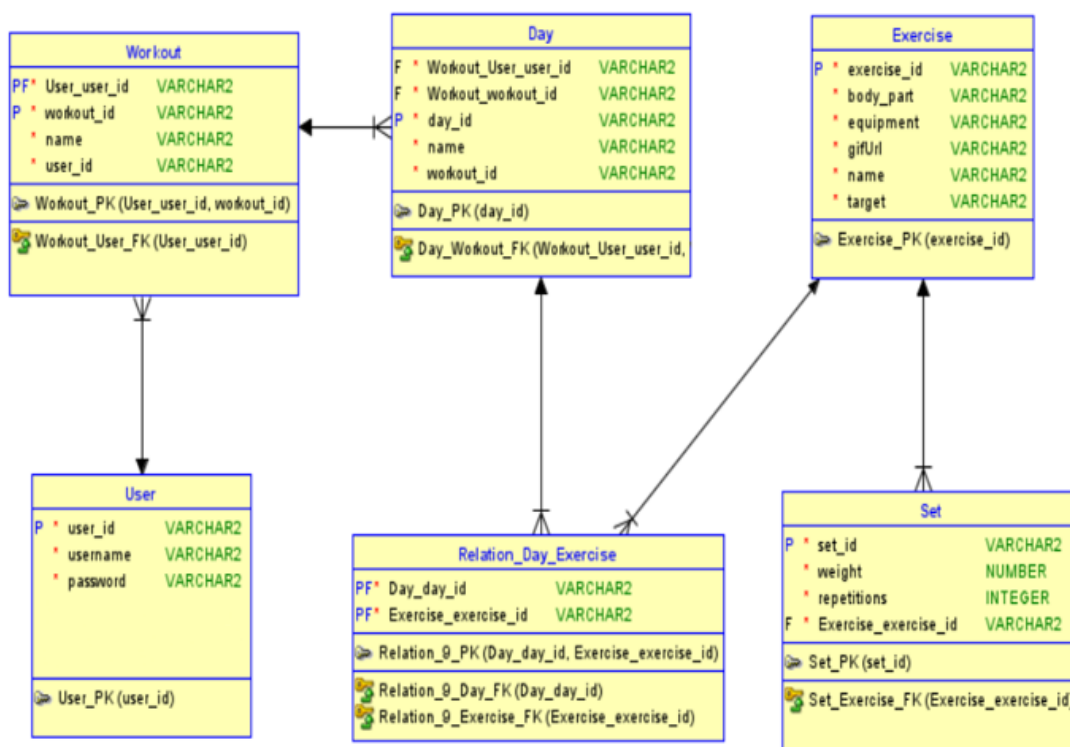


Fig. 5.2.1.2. Model relacional de la base de dades. Font: Elaboració pròpia.

El model relacional permet diferenciar els diversos tipus de variables assignades a cada atribut. La gran majoria d'atributs requereixen combinacions de lletres per tal d'atribuir informació general. D'altra banda, també existeixen atributs que requereixen valors numèrics.

A més, cal destacar l'existència de claus foranes, que estableixen relacions amb claus primàries (atributs identificadors) de les altres taules. Aquests tipus de clau són necessàries per a assegurar la integritat referencial, fet que permet que les dades d'una taula relacionada sempre faci referència a un registre existent de la taula.

5.2.2. Implementació.

Mitjançant l'estructura de la base de dades dissenyada, es duu a terme la implantació de la base de dades i establir una comunicació amb el funcionament de la mateixa aplicació mòbil. La creació de la base de dades s'ha realitzat amb l'ús de JPA, una API de Java que ofereix un conjunt de classes i interfícies per gestionar la persistència d'objectes Java en una base de dades relacional.

JPA permet crear aplicacions que interactuen amb bases de dades relacionals sense haver d'escriure SQL directament. Això es porta a cap mitjançant un model de programació orientat a objectes, on les classes Java són traduïdes a taules de la base de dades, i les instàncies de les classes són convertides a registres dins les taules.

JPA simplifica el desenvolupament d'aplicacions que fan ús de bases de dades relacionals, ja que permet centrar-se en la lògica de negoci de l'aplicació, mentre que l'eina s'encarrega de la gestió de la persistència de les dades. A més a més, JPA ofereix una forma estandarditzada d'interactuar amb les bases de dades, independentment del proveïdor de base de dades utilitzat, fet que implica que les aplicacions siguin més portables i fàcils de mantenir.

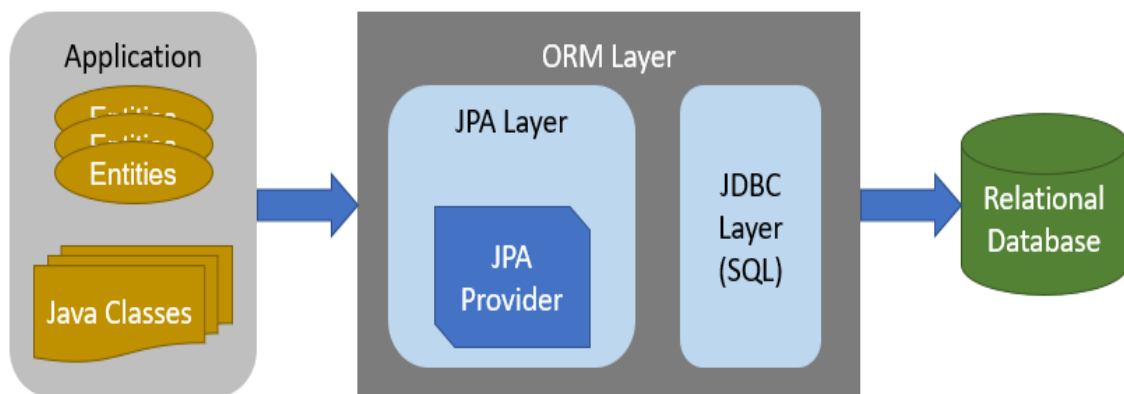


Fig. 5.2.2.1. Estructura per capes del funcionament de JPA. Font:

<https://thistechnologylife.com/object-persistence-with-jpa/>

Tal com es pot observar, l'aplicació es connecta a la capa de JPA mitjançant un proveïdor de JPA, és a dir, implementacions de JPA que utilitzen les seves interfícies per aportar les funcionalitats necessàries. En el context del present TFG, s'usa Hibernate com a implementació.

La conversió es realitza i es connecta a l'API JDBC, que permet connectar-se a una base de dades i executar les peticions corresponents a les traduccions realitzades per JPA i així obtenir el model relacional desitjat.

Per tal de fer efectiva la conversió comentada, JPA permet la utilització d'anotacions dins del codi, amb l'objectiu d'assignar les diverses taules i relacions a generar.

A continuació, es mostra un fragment de la classe *Exercise*, junt amb diverses anotacions necessàries per a la seva conversió a una taula relacional:

```
@Entity
public class Exercise {
    private String bodyPart;
    private String equipment;
    private String gifUrl;
    @Id
    private Long id;
    private String name;
    private String target;
```

Fig. 5.2.2.2. Fragment de la classe *Exercise* amb anotacions de JPA. Font: Elaboració pròpia.

A la figura mostrada es poden observar dues anotacions diferents. Aquestes poden identificar-se mitjançant la localització del símbol “@” a l’inici de cada notació. La primera notació, amb la paraula *Entity*, indica la necessitat de convertir la classe *Exercise* en una entitat, és a dir, una taula relacional. La segona notació, tal com indica la mateixa, assigna un atribut identificador a la taula, que compleix la funció de clau primària, dada necessària per identificar les taules. Aquesta notació s’assigna a una variable de la classe i ha de servir com a forma de reconeixement.

Un altre punt a destacar sobre la implementació de la base de dades és la necessitat de traduir les llistes d’objectes. Algunes classes del Back End utilitzen llistes d’objectes associats a altres classes, fet que cal reflectir a la capa de persistència de dades. L’ús de llistes d’objectes es pot traduir en una relació entre dues entitats diferents. JPA permet l’ús d’anotacions per indicar aquests aspectes més específics.

A continuació, es mostren les anotacions usades per configurar les relacions entre entitats corresponents:

```
@OneToMany(targetEntity = Set.class, cascade = CascadeType.ALL)
@JoinColumn(name = "exercise_set_fk", referencedColumnName = "id")
private List<Set> sets = new LinkedList<Set>();
```

Fig. 5.2.2.3. Anotacions de JPA per establir la relació entre les entitats *Exersice* i *Set*. Font: Elaboració pròpia.

La primera notació designa el tipus de relació entre les dues entitats esmentades. En aquest cas, es pot observar que la relació establerta correspon al tipus 1:M. La llista mostrada a la figura conté objectes de la classe *Set*, per tant, la relació indica que un objecte de la classe *Exercise* estigui relacionat amb diversos objectes de la classe *Set*. El paràmetre *CascadeType.ALL* permet establir una regla de cascada per a la relació entre les dues taules. Això significa que si s'efectua una operació de modificació o eliminació a la fila principal de la relació (en aquest cas, la taula *Exercise*), les operacions corresponents es propaguen a les files relacionades a la taula *Set*. Aquesta funció permet mantenir la integritat de les dades.

Per altra banda, la notació *@JoinColumn* permet establir una associació entre dues taules en una base de dades relacional. Aquesta anotació s'utilitza per indicar quina columna de la taula propietària de l'objecte ha de ser utilitzada com a clau forana per establir la relació amb la taula objectiva. En aquest cas, la columna usada com a clau forana (anomenada *exercise_set_fk*) és la variable *id*, corresponent a la variable identificadora de l'objecte i, per conseqüència, de la taula.

Per tal d'obtenir la base de dades especificada mitjançant les diverses notacions de JPA arreu del codi, s'ha utilitzat XAMPP, un paquet de programari que inclou diversos components destinats per a la creació i gestió d'aplicacions. Dins d'aquests components, inclou un servidor web Apache i una base de dades amb el sistema gestor MySQL, el qual usa el llenguatge SQL. El servidor web Apache permet l'execució de l'aplicació en un entorn local, fet que permet un desenvolupament i desplegament segur, mentre que el sistema gestor de bases de dades MySQL crea la base de dades en funció de les especificacions assignades al codi mitjançant JPA, tal com s'ha esmentat anteriorment.

5.3. Aplicació mòbil.

En el context de la mateixa aplicació, es diferencien clarament dos grans blocs, el Back End i el Front End, el procés de desenvolupament dels quals es descriu de forma separada i específica.

5.3.1. Disseny.

En la secció del Back End, per tal d'organitzar el codi de forma estructurada i modular, facilitar el seu manteniment, escalabilitat i reutilització, s'ha portat a terme el disseny d'un esquema de paquets del projecte.

El disseny d'aquest esquema ajuda a separar les diferents parts del codi en mòduls independents, la qual cosa facilita l'enteniment de les funcionalitats del codi i ajuda a evitar conflictes entre diferents àrees d'aquest. A més, permet una millor gestió de les dependències, el que, a la vegada, millora la qualitat del codi i redueix el temps de desenvolupament. Addicionalment, aquesta tasca també és important per garantir la seguretat i l'eficiència del codi, ja que permet aplicar diferents nivells de permisos i accés als diferents mòduls del projecte.

A continuació, es mostra el disseny de l'esquema de paquets que conformen el Back End de l'aplicació, junt amb l'addició de les classes existents a cada paquet relacionades amb l'entitat *Exercise*, per tal de contextualitzar l'organització mostrada:

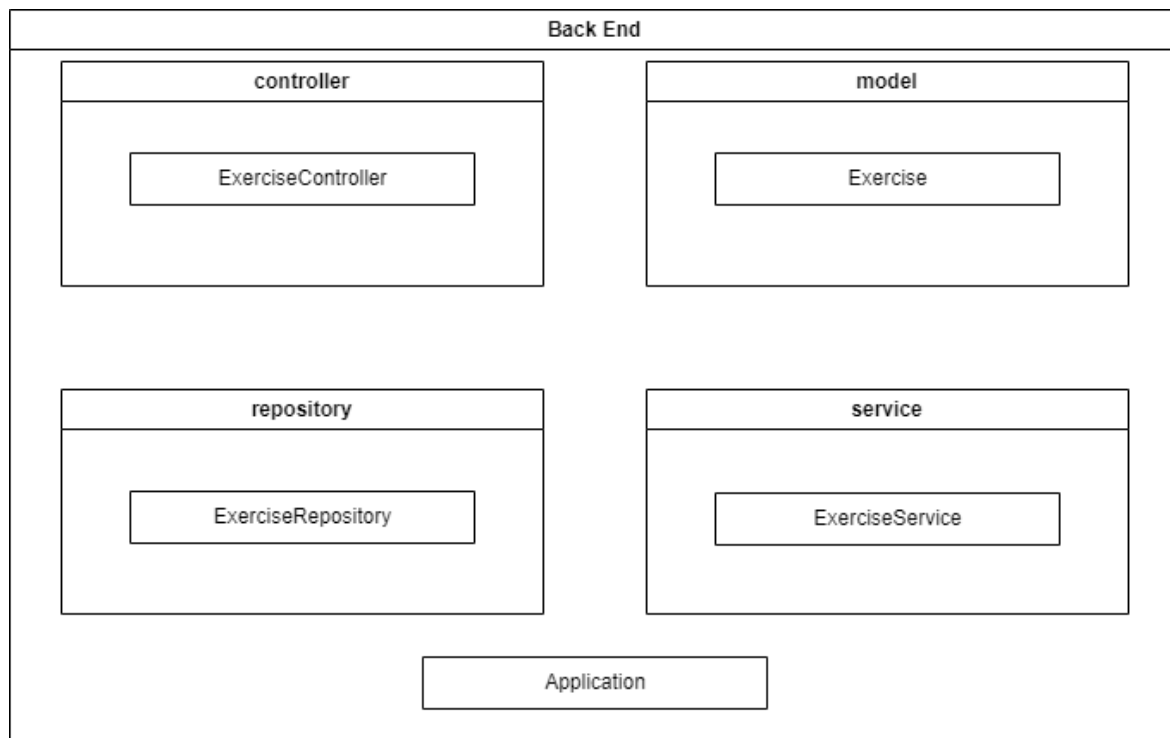


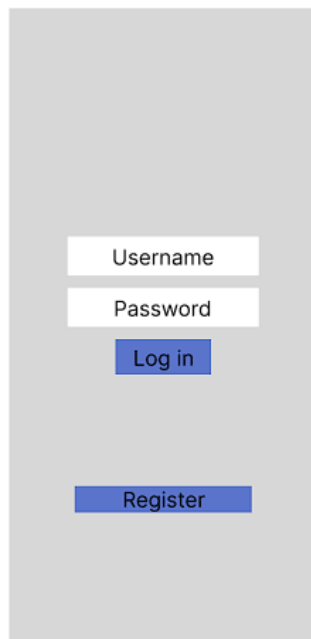
Fig. 5.3.1.1. Esquema de paquets del Back End. Font: Elaboració pròpia.

L'estructura del diagrama mostrat segueix el model de l'arquitectura de capes. Aquesta arquitectura es caracteritza per la separació per capes, tal com el seu nom indica, en les que cadascuna d'aquestes té atribuïda una responsabilitat específica.

En el context d'aquest projecte, el paquet *controller* és l'encarregat de gestionar les peticions HTTP i les respostes obtingudes del servidor. El paquet anomenat *model* té la responsabilitat de definir els objectes de negoci de l'aplicació. El paquet *repository* permet gestionar les operacions de la base de dades i, finalment, el paquet *service* s'encarrega de la lògica de negoci i de la comunicació amb la capa de persistència (paquet *repository*).

En l'àmbit del Front End de l'aplicació s'han dissenyat diverses interfícies que mostren l'estructura bàsica dels components i funcionalitats de cadascuna.

A continuació, es procedeix a mostrar i descriure les interfícies principals de l'aplicació:



The image shows a vertical grey rectangle representing a mobile app screen. In the center, there are two white input fields stacked vertically. The top field is labeled 'Username' and the bottom field is labeled 'Password'. Below these fields is a blue button with the text 'Log in'. Further down, there is another blue button with the text 'Register'.

Fig. 5.3.1.2. Disseny de la interfície d'autenticació d'usuari. Font: Elaboració pròpia.

La interfície d'autenticació d'usuari permet registrar-se o bé autenticar-se, en funció de si l'usuari en qüestió accedeix a l'aplicació per primer cop o no.

La interfície ofereix camps per introduir el nom d'usuari i contrasenya, a més dels botons d'autenticar-se i registrar-se, tal com s'ha mencionat anteriorment.



The image shows a vertical grey rectangle representing a mobile app screen. At the top, there is a white header bar with the text 'Workout list'. Below this, there is a table with four rows. Each row contains a workout number and the number of days. At the bottom of the screen, there is a blue circular button with a white plus sign inside.

| Workout list | |
|--------------|--------|
| Workout #1 | 4 days |
| Workout #2 | 3 days |
| Workout #3 | 5 days |
| Workout #4 | 6 days |

Fig. 5.3.1.3. Disseny de la interfície d'entrenaments. Font: Elaboració pròpia.

Aquesta interfície es mostra després que un usuari realitzi l'autenticació i permet visualitzar les diverses rutines d'entrenament creades pel mateix usuari, en cas d'haver-ne creat prèviament. A la secció inferior de la interfície apareix el botó que permet crear una nova rutina d'entrenament.



Fig. 5.3.1.4. Disseny de la interfície de dies d'entrenament. Font: Elaboració pròpia.

La interfície mostrada es mostra quan l'usuari escull una rutina d'entrenament. De la mateixa manera que a la interfície anterior, si l'usuari no ha afegit cap dia d'entrenament a la rutina, la llista de dies apareix buida i cal que insereixi dies nous mitjançant el botó ubicat a la secció inferior de la interfície.

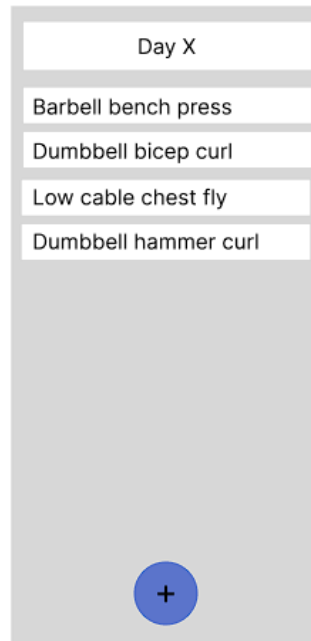


Fig. 5.3.1.5. Disseny de la interfície d'exercicis d'un dia. Font: Elaboració pròpia.

La interfície d'exercicis mostra la llista d'exercicis a realitzar durant un dia d'entrenament. Tal com s'ha esmentat prèviament, la llista d'exercicis pot ser modificada en funció de les preferències d'usuari.

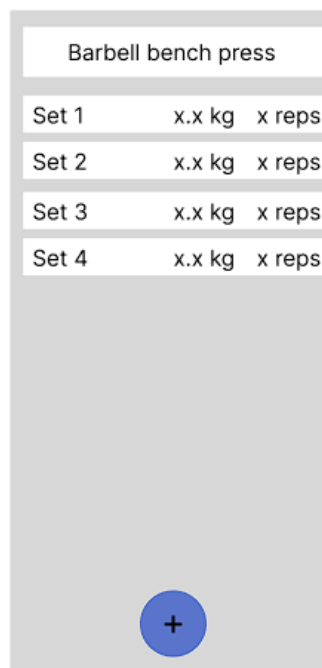


Fig. 5.3.1.6. Disseny de la interfície de sèries d'un exercici. Font: Elaboració pròpia.

La interfície mostrada mostra les característiques d'un exercici amb relació a la informació de cada sèrie. L'usuari pot afegir noves sèries i establir el pes utilitzat com a càrrega i les repeticions realitzades per cadascuna d'aquestes.

5.3.2. Implementació.

Un cop descrits els requisits i tasques de disseny de cada aspecte per aconseguir l'assoliment dels objectius del present TFG, es procedeix en aquest apartat a definir realitzar la implementació real i funcional del Back End i Front End de l'aplicació mòbil. Cal destacar que la descripció no aprofundeix en tota la implementació del codi línia per línia, ja que es vol mostrar únicament els punts més importants i destacables.

En la secció del Back End de l'aplicació, cal destacar la necessitat de l'ús d'una API externa que ofereix una gran quantitat d'exercicis per a fer, els quals són carregats a la base de dades. L'objectiu d'aquesta acció es basa a poder oferir a l'usuari un gran repertori d'exercicis per escollir a l'hora de crear la seva pròpia rutina d'entrenament.


```
@RestController
@RequestMapping("/exercises")
public class ExerciseController {

    private static final String BASE_URL = "https://exercisedb.p.rapidapi.com/exercises";
    private static final String API_KEY = "d3284ac5f2msh38f86ca49aa9a46p140e76jsn1e7c29c446b4";
    private static final String API_HOST = "exercisedb.p.rapidapi.com";

    private final ObjectMapper objectMapper = new ObjectMapper();

    @Autowired
    private ExerciseService exerciseService;

    @PostMapping("/saveAll")
    public void saveExercises() {
        try {
            HttpResponse<String> response = Unirest.get(BASE_URL)
                .header("X-RapidAPI-Key", API_KEY)
                .header("X-RapidAPI-Host", API_HOST)
                .asString();
            String responseBody = response.getBody();
            List<Exercise> exercises = objectMapper.readValue(responseBody, new TypeReference<List<Exercise>>() {});
            exerciseService.saveExercises(exercises);
        } catch (Exception e) {}
    }
}
```

Fig. 5.3.2.1. Codi de la funció de recopilació i emmagatzematge d'exercicis de l'API externa. Font: Elaboració pròpia.

El fragment de codi mostrat pertany a la classe *ExerciseController*, la qual gestiona totes les peticions HTTP relacionades amb l'entitat *Exercise*. Tal com es pot observar a la figura mostrada, la funció *saveExercises* s'encarrega d'enviar una petició a l'API externa mitjançant un URL, clau i *host* per obtenir una resposta que conté una àmplia llista d'exercicis, la qual és estandarditzada (mitjançant la instància *objectMapper*) per tal d'emmagatzemar una llista d'objectes de la classe *Exercise* a la base de dades.

Un cop s'ha efectuat la càrrega de dades relacionada amb l'API externa, totes les següents peticions a realitzar són de caràcter local i ofereixen totes les funcionalitats disponibles mitjançant una API pròpia. Mitjançant aquesta API local, el Front End realitza les comunicacions corresponents per a poder realitzar les accions que requereixen interacció amb el Back End i la base de dades.

A continuació, es procedeix a llistar les funcionalitats que ofereix la infraestructura Back End de l'aplicació mòbil:

- Creació i eliminació de rutines d'entrenament.
- Creació i eliminació de dies d'entrenament.
- Creació i eliminació d'exercicis.
- Creació, modificació i eliminació de sèries d'exercicis.

A més, cal destacar l'ús de Spring Security, un entorn de control d'accés i autenticació que ofereix Spring Boot com a mòdul per a implementar a les aplicacions desenvolupades amb aquest entorn. El mòdul conté diverses característiques i eines que permeten aplicar les mesures de protecció i seguretat corresponents en funció de les necessitats de cada aplicació concreta. En el cas del present TFG, s'han implementat les funcionalitats d'autenticació d'usuaris i registre de nous usuaris. Concretament, la metodologia d'implementació utilitzada es basa en el xifratge i verificació de claus d'accés mitjançant la classe *BCryptPasswordEncoder*. Aquesta classe de Spring Security utilitza l'algorisme *bcrypt* com a mètode d'enciptació per tal d'emmagatzemar contrasenyes enciptades a la base de dades amb l'objectiu de protegir les credencials de cada usuari.

Mitjançant un objecte de la classe *BCryptPasswordEncoder*, s'encipten les claus d'accés dels usuaris registrats i es comparen les claus d'accés dels usuaris que volen iniciar sessió. A continuació, es mostren les funcions d'inici de sessió d'un usuari i de registre d'un nou usuari de la classe *UserService*:

```
1 usage
public boolean loginUser(User user) {
    User existingUser = userRepository.findByUsername(user.getUsername());
    if (existingUser != null) {
        if (user.getUsername().equals(existingUser.getUsername()) && passwordEncoder.matches(user.getPassword(), existingUser.getPassword())) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

1 usage
public boolean registerUser(User user) {
    if (userRepository.existsByUsername(user.getUsername())) {
        return false;
    } else {
        user.setPassword(passwordEncoder.encode(user.getPassword()));
        userRepository.save(user);
        return true;
    }
}
```

Fig. 5.3.2.2. Codi de les funcions d'inici de sessió i de registre d'usuari. Font: Elaboració pròpia.

Tal com es pot observar a la figura, l'autenticació d'usuari es realitza mitjançant la comparació d'una contrasenya sense xifrar amb xifrada per tal d'assegurar que són la mateixa clau. En el cas de la funció de registre d'un nou usuari, la contrasenya s'encrypta mitjançant l'objecte *passwordEncoder* de la classe *BCryptPasswordEncoder* abans de ser emmagatzemada a la base de dades.

A part de la configuració de seguretat descrita, ha estat necessària la implementació de la següent classe:

```
@Configuration
@EnableWebMvc
public class WebConfig implements WebMvcConfigurer {

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping(pathPattern: "**")
            .allowedMethods("GET", "POST", "DELETE", "PUT");
    }
}
```

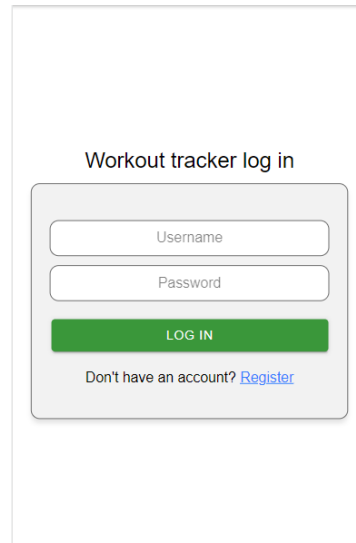
Fig. 5.3.2.3. Codi de la classe *WebConfig*. Font: Elaboració pròpia.

Aquesta classe permet habilitar el mecanisme de seguretat CORS (Cross-Origin Resource Sharing), implementat als navegadors web que permet a un servidor indicar als navegadors si han de permetre sol·licituds de recursos des d'un origen diferent (domini o port).

Com que la infraestructura Back End es troba en un altre domini, ha estat necessària la implementació de la classe mostrada a la figura. Concretament, la classe *WebConfig* amb l'anotació *@Configuration* i la implementació de la interfície *WebMvcConfigurer* és necessària per configurar la funcionalitat de CORS en una aplicació desenvolupada mitjançant Spring Boot.

Per altra banda, la secció del Front End ha estat desenvolupada mitjançant la implementació de *Components*. Els *Components* són elements d'interfície d'usuari que ofereix Ionic React que formen els blocs de construcció fonamentals per poder crear interfícies d'usuari per aplicacions web i mòbil. Els *Components* ofereixen una àmplia gamma de funcionalitats i característiques a adaptar-se als requeriments de cada aplicació.

A continuació, es procedeix a mostrar el conjunt d'interfícies desenvolupades, junt amb les diverses funcionalitats que ofereixen:



Workout tracker log in

Username

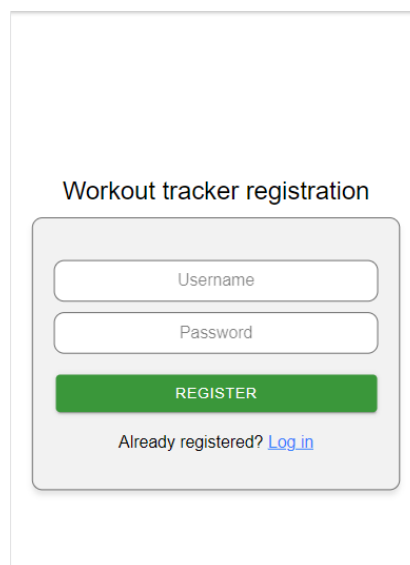
Password

LOG IN

Don't have an account? [Register](#)

Fig. 5.3.2.4. Interfície d'inici de sessió. Font: Elaboració pròpia.

Aquesta interfície és el primer element que es visualitza quan s'executa l'aplicació i ofereix un formulari per poder introduir el nom d'usuari i contrasenya per tal que un usuari pugui iniciar sessió. El formulari conté una funcionalitat que permet verificar les credencials mitjançant comunicació amb el Back End, el qual verifica si les credencials d'un usuari són correctes o si existeixen a la base de dades. En funció del resultat, s'inicia la sessió de l'usuari o es notifica un error. A més, l'usuari pot accedir a la interfície de registre en cas de no tenir un compte d'usuari registrat.



Workout tracker registration

Username

Password

REGISTER

Already registered? [Log in](#)

Fig. 5.3.2.5. Interfície de registre d'usuari. Font: Elaboració pròpia.

Aquesta interfície permet a l'usuari registrar un nou compte per poder accedir a l'aplicació. Tal com succeeix al formulari d'inici de sessió, el formulari de registre es comunica amb el Back End per verificar la creació del compte d'usuari. En cas d'error, es mostra una notificació. Addicionalment, es pot accedir a la interfície d'inici de sessió per tal d'obtenir accés a l'aplicació.

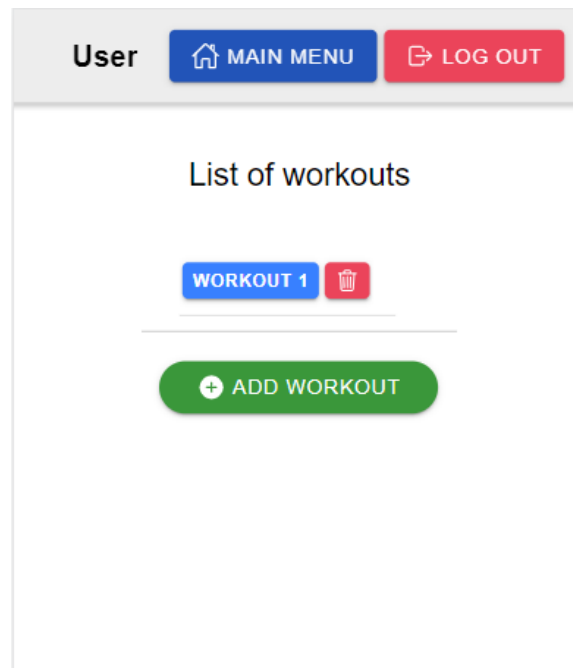
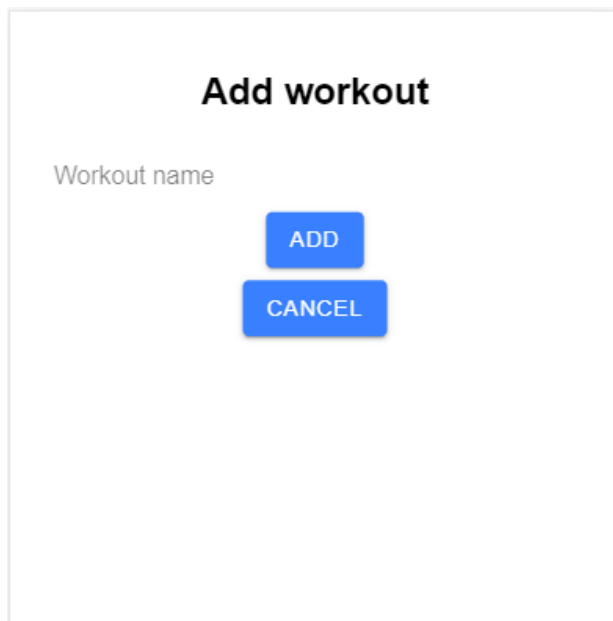


Fig. 5.3.2.6. Interfície de llista de rutines d'un usuari. Font: Elaboració pròpia.

La interfície de llista de rutines és la pàgina principal de l'aplicació i mostra les diverses rutines creades per part d'un usuari. L'usuari pot eliminar les rutines ja existents o bé crear noves rutines mitjançant el botó corresponent.

Per altra banda, cal destacar que les interfícies de l'aplicació, un cop un usuari hagi iniciat sessió, ofereixen les funcionalitats de tornar al menú principal (redirigeix a aquesta interfície).

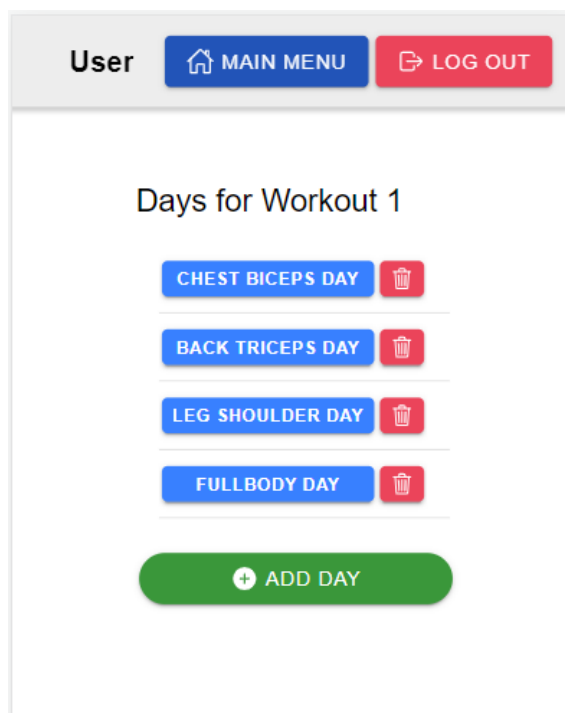
En cas de crear una nova rutina, es mostra la interfície de creació d'una rutina, mostrada a continuació:



The screenshot shows a form titled "Add workout". Below the title is a text input field labeled "Workout name". Underneath the input field are two blue buttons: "ADD" and "CANCEL".

Fig. 5.3.2.7. Interfície de creació d'una nova rutina. Font: Elaboració pròpia.

Tal com s'ha esmentat prèviament, aquesta interfície mostra un formulari per crear una nova rutina d'exercici. L'usuari ha d'introduir un nom per poder identificar la rutina per tal que el Front End pugui comunicar-se amb l'API del Back End i així establir la creació d'una nova rutina.



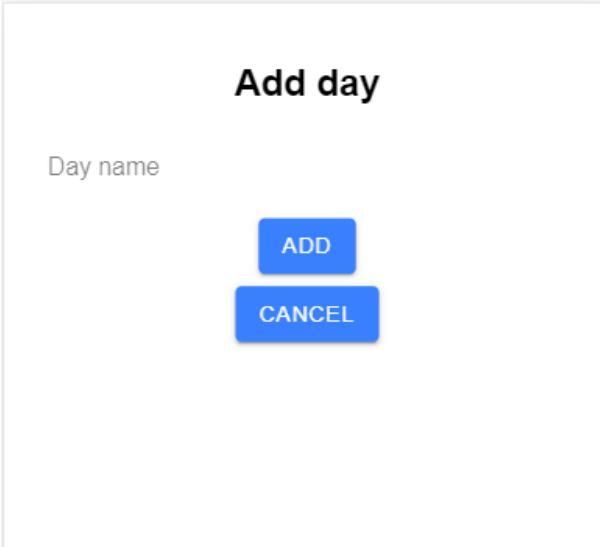
The screenshot shows a user interface for a workout routine. At the top, there is a header with the word "User" and two buttons: "MAIN MENU" (with a home icon) and "LOG OUT" (with a door icon). Below the header, the title "Days for Workout 1" is displayed. Underneath, there is a list of four workout days, each with a blue button and a red trash icon:

- CHEST BICEPS DAY
- BACK TRICEPS DAY
- LEG SHOULDER DAY
- FULLBODY DAY

At the bottom of the list, there is a green button with a plus sign and the text "ADD DAY".

Fig. 5.3.2.8. Interfície de llista de dies d'una rutina. Font: Elaboració pròpia.

Mitjançant una rutina d'exercici, l'usuari pot accedir a la llista de dies assignada a la rutina corresponent. De la mateixa manera que a la llista de rutines, l'usuari pot crear nous dies d'entrenament i també eliminar-ne. En cas de crear un nou dia, es mostra el següent formulari:



The image shows a mobile application interface for adding a new day. It features a white background with a thin grey border. At the top center, the text "Add day" is displayed in a bold, black font. Below this, the label "Day name" is positioned to the left of a text input field. At the bottom center, there are two blue buttons with white text: "ADD" and "CANCEL".

Fig. 5.3.2.9. Interfície de creació d'un nou dia. Font: Elaboració pròpia.

Aquesta interfície té la funció de formulari per a la creació de nous dies d'entrenament. Quan l'usuari introdueix un nom per assignar al dia i afegeix el dia, es mostra a la llista de dies mitjançant comunicacions amb l'API del Back End.

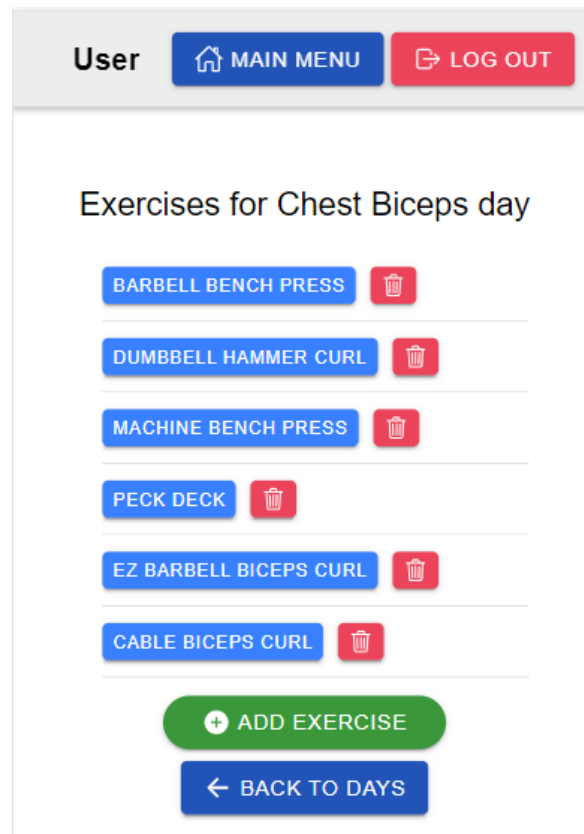


Fig. 5.3.2.10. Interfície de llista d'exercicis d'un dia. Font: Elaboració pròpia.

La interfície d'exercicis es mostra quan s'accedeix a un dia en concret de la llista de dies. Aquesta llista mostra els exercicis assignats a un dia i permet crear nous exercicis i esborrar exercicis existents. A més de la funcionalitat de creació d'un nou exercici que desplega un formulari, la interfície mostra un botó per poder tornar a la llista de dies.

Add exercise

Exercise name

ADD

CANCEL

Existing exercises

Search exercises

dumbbell alternate seated hammer curl

dumbbell alternate side press

dumbbell alternating bicep curl with leg raised o...

dumbbell arnold press v. 2

dumbbell around pullover

dumbbell bench press

Fig. 5.3.2.11. Interfície de creació d'un nou exercici. Font: Elaboració pròpia.

Aquesta interfície permet assignar un nou exercici al dia seleccionat i ofereix dues opcions per realitzar-ho: Creació d'un exercici nou i selecció d'un exercici existent. La creació d'un exercici nou permet assignar qualsevol nom a l'exercici, mentre que la selecció d'un exercici existent permet assignar un nou exercici amb el nom escollit de la llista d'exercicis existents. Aquesta llista mostra tots els exercicis existents a la base de dades mitjançant la comunicació amb l'API del Back End.

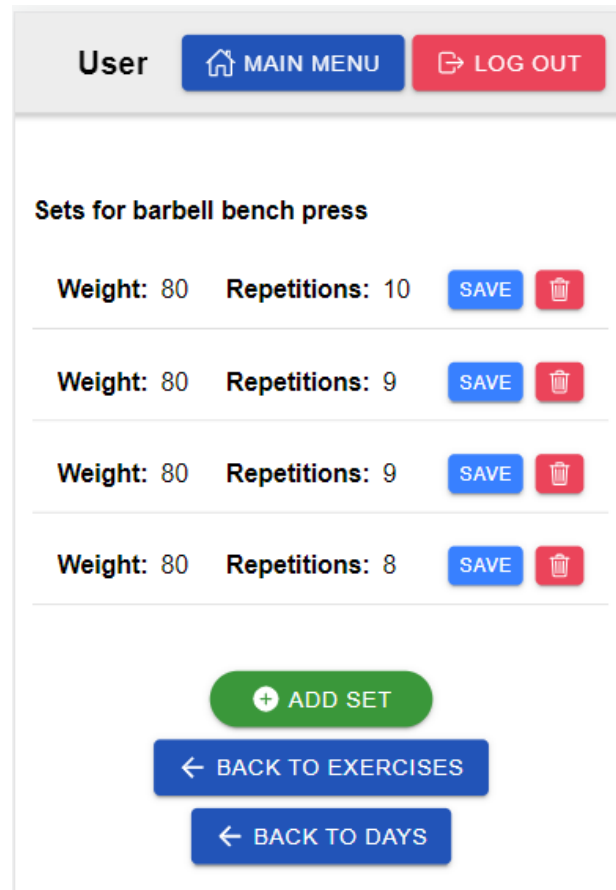
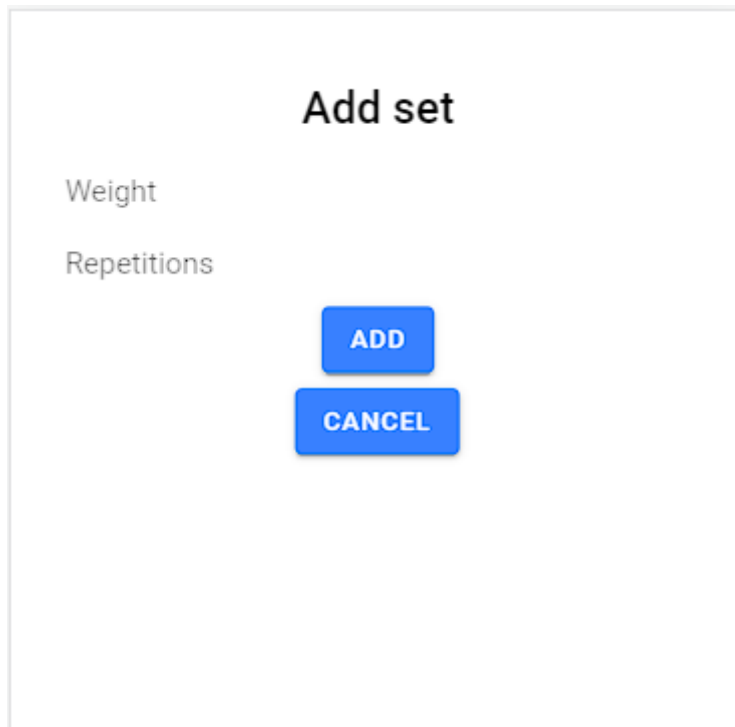


Fig. 5.3.2.12. Interfície de llista de sèries d'un exercici. Font: Elaboració pròpia.

La interfície mostrada permet visualitzar les sèries assignades a un exercici en concret. Aquesta interfície permet crear, modificar i eliminar sèries, a part de poder tornar a visualitzar la llista d'exercicis i dies. Si l'usuari vol crear una nova sèrie, es mostra la següent interfície:



The image shows a mobile application interface for adding a new set. At the top, the title "Add set" is centered. Below the title, there are two input fields: "Weight" and "Repetitions". At the bottom of the form, there are two blue buttons: "ADD" and "CANCEL".

Fig. 5.3.2.13. Interfície de creació d'una nova sèrie. Font: Elaboració pròpia.

Finalment, aquesta interfície permet crear una nova sèrie per a un exercici. Com es pot observar, el formulari demana introduir la càrrega i les repeticions realitzades.

6. Possibles ampliacions.

El projecte ha assolit el seu objectiu principal i ha mostrat correspondència amb els requeriments descrits durant la fase inicial del TFG. No obstant, durant el seu desenvolupament han sorgit noves idees, necessitats i objectius que poden dur-se a terme en un futur. En aquest apartat es presenten possibles addicions per al projecte, amb l'objectiu de millorar i expandir les seves funcionalitats.

A continuació, es llisten els possibles avanços que es poden realitzar en termes d'ampliacions futures:

- **Localització:** Encara que l'aplicació mostra el text en anglès, introduir la possibilitat d'escollir l'idioma pot facilitar el seu ús arreu dels diversos països i regions.
- **Visualització de contingut visual dels exercicis:** Els exercicis escollits poden mostrar vídeos o contingut en format GIF per tal de mostrar la tècnica realització d'un exercici.
- **Recomanacions d'exercicis:** Incorporar algorismes i tècniques d'intel·ligència artificial per oferir als usuaris una selecció d'exercicis a escollir en funció dels exercicis escollits prèviament.
- **Assessorament de rutines:** Oferir un formulari amb els paràmetres necessaris per poder crear automàticament una rutina d'exercici per a un usuari que no tingui una rutina d'exercici prèvia o que no posseeixi experiència en l'àmbit de l'exercici.
- **Personalització de la interfície d'usuari:** Brindar als usuaris la capacitat de modificar els paràmetres de les diverses interfícies de l'aplicació en funció de les seves preferències i necessitats. Alguns paràmetres modificables inclouen el color del tema, la disposició d'elements i la mida de la font.
- **Hosting del Back End i la base de dades:** Allotjar l'execució del Back End i la base de dades a un servidor extern per tal de mantenir la infraestructura sempre operativa i en una base de dades comuna entre tots els usuaris existents.

En resum, les possibles ampliacions descrites ofereixen un conjunt d'oportunitats per enriquir i millorar el projecte en termes de funcionalitats, usabilitat i adaptació a les necessitats dels usuaris. La incorporació d'aquestes ampliacions permet el constant impuls,

creixement i innovació del projecte, brindant una experiència més completa i satisfactòria als usuaris.

7. Anàlisi de resultats i conclusions.

En aquest apartat es procedeix a examinar els resultats obtinguts mitjançant la realització del projecte en relació amb els objectius plantejats durant la fase inicial del TFG.

Concretament, es fa referència en la metodologia seguida i els efectes sobre la realització del projecte i en les tecnologies emprades per assolir la solució final.

7.1. Resultats de la metodologia.

L'ús de la metodologia àgil en aquest TFG ha demostrat ser beneficiós en diversos aspectes. A continuació, es presenta una anàlisi dels resultats obtinguts mitjançant aquest mètode:

- **Flexibilitat:** La metodologia àgil ha permès adaptar-se i respondre de manera efectiva als canvis i demandes constants del projecte. Mitjançant la planificació iterativa s'han pogut realitzar canvis i addicions constants durant tot el desenvolupament del projecte.
- **Lliurament incremental:** El lliurament regular d'addicions funcionals al programari (per cada lliurament s'entén la finalització d'una iteració) ha permès obtenir comentaris primerencs dels usuaris i realitzar modificacions en funció de les seves necessitats i preferències.
- **Comunicació eficaç:** Aquesta metodologia ha emfatitzat en la comunicació i col·laboració entre l'alumne i tutor del present TFG. Les reunions periòdiques han assegurat un correcte progrés del projecte i un desenvolupament viable.
- **Adaptabilitat i aprenentatge:** La metodologia àgil ha permès la millora contínua i l'aprenentatge al llarg del projecte. Mitjançant la retroalimentació continuada i l'avaluació de resultats, s'han pogut identificar diverses oportunitats de millora i implementar solucions per optimitzar el desenvolupament i els resultats finals.

En resum, la metodologia àgil ha aportat beneficis significatius en tots els aspectes mencionats prèviament, resultant en una millora important del procés de desenvolupament del present TFG. Amb això, els reptes del projecte han pogut ser afrontats de manera efectiva, fet que ha comportat resultats satisfactoris en termes de la realització del projecte.

7.2. Resultats de les tecnologies.

La realització d'aquest TFG ha permès demostrar l'efectivitat i la utilitat de les tecnologies utilitzades, com XAMPP com a base de dades, Spring Boot com a Back End i Ionic React per al Front End. A continuació, s'analitzen els resultats i les conclusions obtinguts mitjançant la implementació d'aquestes tecnologies.

L'ús de XAMPP com a base de dades ha demostrat oferir una solució robusta, fiable per a l'emmagatzematge i la gestió de les dades del sistema mitjançant l'ús d'un servidor web Apache i el sistema gestor de bases de dades MySQL que ofereix l'eina. La configuració i administració de la base de dades s'ha realitzat de manera senzilla i local, cosa que ha permès un ràpid desenvolupament i desplegament del sistema gestor.

D'altra banda, la utilització de Spring Boot com per al desenvolupament del Back End ha brindat un marc de treball poderós i ben estructurat per desenvolupar aplicacions web. La facilitat de configuració i l'àmplia gamma de funcionalitats integrades, com ara la gestió de la seguretat i la implementació d'APIs, han permès una ràpida implementació de les característiques requerides per l'aplicació. A més, la integració de Spring Boot amb XAMPP ha estat fluida mitjançant l'ús de JPA, fet que ha facilitat la comunicació entre el Back End i la base de dades.

Pel que fa a l'ús de Ionic React com entorn de desenvolupament del Front End, s'ha aconseguit desenvolupar una interfície d'usuari d'acord amb els criteris d'experiència d'usuari requerits. Ionic React ofereix un conjunt complet de *Components* i eines de disseny que faciliten la creació d'interfícies intuïtives. La integració amb React ha permès un desenvolupament àgil i escalable, aprofitant els avantatges de la programació basada en *Components*. A més, Ionic React ha permès desenvolupar una aplicació compatible amb dispositius mòbils i ordinadors.

Finalment, l'ús de XAMPP, Spring Boot i Ionic React en aquest projecte ha permès obtenir resultats satisfactoris en termes de rendiment, funcionalitat i usabilitat. Aquestes tecnologies han demostrat ser una combinació sòlida i eficient per desenvolupar aplicacions mòbils híbrides. El coneixement adquirit durant la implementació d'aquest projecte estableix les bases per al desenvolupament de futurs sistemes basats en aquestes tecnologies, amb el potencial d'oferir solucions d'alta qualitat i valor afegit.

7.3. Conclusions.

En aquest TFG s'ha desenvolupat i presentat un projecte que té com a objectiu oferir una plataforma de registre i seguiment de rutines d'exercici per a usuaris interessats a mantenir un estil de vida actiu i progressar en les càrregues utilitzades per a la realització dels exercicis.

Durant aquest el procés de realització del projecte, s'ha aplicat la metodologia àgil, la qual ha permès una gestió eficient del projecte i una adaptació continua a les necessitats i requisits dels usuaris, tal com s'ha esmentat anteriorment.

Al llarg del desenvolupament del present TFG, s'ha emfatitzat en la usabilitat, accessibilitat i funcionalitat de l'aplicació, buscant proporcionar una experiència d'usuari intuïtiva i satisfactòria. S'han implementat diverses característiques, com la filtració d'exercicis per nom per poder indagar entre la llista d'exercicis que ofereix de forma predeterminada l'aplicació i el registre i actualització de càrrega i repeticions realitzades durant una sèrie d'un exercici.

De la mateixa manera, s'ha elaborat una anàlisi dels tests d'usuari, verificant la usabilitat, accessibilitat i funcionalitat de l'aplicació (descriu en profunditat a l'estudi de viabilitat).

Respecte les possibles ampliacions, les idees plantejades tenen l'objectiu d'enriquir el projecte i adaptar-lo a les tendències i necessitats actuals. Aquestes ampliacions contemplen aspectes com la personalització de l'experiència d'usuari, visualització de contingut visual de cada exercici i la localització de l'aplicació en diferents idiomes.

En conclusió, aquest Treball de Final de Grau ha permès desenvolupar un projecte que compleix amb els objectius plantejats inicialment. A més, s'ha posat en pràctica la metodologia de treball àgil, fet que ha facilitat la gestió del projecte i adaptació a les modificacions dels requeriments dels usuaris. Addicionalment, aquest TFG presenta un marge de possibles ampliacions que ofereixen noves oportunitats per expandir i millorar el projecte en el futur, ja que presenta una base escalable alhora que actualment l'aplicació és totalment funcional i ha complert amb els objectius i requisits plantejats. En definitiva, l'aplicació desenvolupada ofereix una experiència d'usuari satisfactòria i proporciona eines útils per al seguiment i registre de rutines d'exercici acord amb els objectius establerts.

8. Bibliografia.

- [1] LBDC and elPeriodico.com, “¿Qué son series y repeticiones en una rutina de entrenamiento?,” La Bolsa del Corredor, Oct. 07, 2021. [en línea] [consulta: 4 de gener de 2023]. Disponibilitat: <https://www.sport.es/labolsadelcorredor/series-y-repeticiones-entrenamiento/>
- [2] J. W. Krieger, “Single vs. Multiple Sets of Resistance Exercise for Muscle Hypertrophy: A Meta-Analysis,” Journal of Strength and Conditioning Research, vol. 24, no. 4, pp. 1150–1159, Abr. 2010, doi: 10.1519/jsc.0b013e3181d4d436
- [3] R. Garcia, “LLEGAR AL FALLO MUSCULAR PARA HIPERTROFIA ¿ES NECESARIO?,” Roberto Garcia, Set. 26, 2020. [en línea] [consulta: 4 de gener de 2023]. Disponibilitat: <https://robertothewise.com/llegar-al-fallo-muscular-para-hipertrofia-es-necesario/>
- [4] D. D. Gil, “Fallo muscular, ¿sí o no?,” Vitónica, Oct. 31, 2016. [en línea] [consulta: 5 de gener de 2023]. Disponibilitat: <https://www.vitonica.com/musculacion/fallo-muscular-si-o-no>
- [5] E. D. L. Vega, “Qué es el principio de sobrecarga progresiva y cómo lograrlo?,” FullMusculo, Jun. 15, 2021. [en línea] [consulta: 4 de gener de 2023]. Disponibilitat: <https://fullmusculo.com/sobrecarga-progresiva/>
- [6] Felisa, “Qué es el Framework Spring y las ventajas de utilizarlo,” ▷ Cursos de Programación de 0 a Experto © Garantizados, Dec. 15, 2019. [en línea] [consulta: 3 de gener de 2023]. Disponibilitat: https://unipython.com/que-es-el-framework-spring-y-las-ventajas-de-utilizarlo/?utm_content=cmp-true
- [7] “Por qué DEBES usar Spring Boot en tus proyectos JAVA - //Arteco,” www.arteco-consulting.com. [en línea] [consulta: 4 de gener de 2023]. Disponibilitat: <https://www.arteco-consulting.com/post/por-que-debes-usar-spring-boot>
- [8] M. Benito, “Què és Node.js i quins avantatges té utilitzar aquesta tecnologia,” FP Online, Mar. 21, 2022. [en línea] [consulta: 3 de gener de 2023]. Disponibilitat:

<https://fp.uoc.fje.edu/ca/blog/que-es-node-js-i-quins-avantatges-te-utilitzar-aquesta-tecnologia/>

[9] M. Lynch, “Ionic React vs React Native: What is The Difference? | Ionic.io,” ionic.io, Feb. 18, 2020. [en línia] [consulta: 3 de gener de 2023]. Disponibilitat: <https://ionic.io/resources/articles/ionic-react-vs-react-native#h-what-is-ionic-react>

[10] A. Del Sole, Visual Studio Code Distilled. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-6901-5

[11] B. F. Tavares et al., “Mobile Applications for Training Plan Using Android Devices: A Systematic Review and a Taxonomy Proposal,” Information, vol. 11, no. 7, p. 343, Jul. 2020, doi: 10.3390/info11070343

[12] AppMaster, “Nuestra guía de las mejores herramientas de desarrollo de aplicaciones móviles (Android / iOS) en 2022 | AppMaster,” appmaster.io, Jun. 17, 2022. [en línia] [consulta: 28 de diciembre de 2022]. Disponibilitat: <https://appmaster.io/es/blog/nuestra-guia-de-las-mejores-herramientas-de-desarrollo-de-aplicaciones-moviles-android-ios-en-2022>

[13] admin, “Los mejores frameworks para desarrollo de apps 2022,” Bambu Mobile, Oct. 11, 2022. [en línia] [consulta: 28 de diciembre de 2022]. Disponibilitat: <https://www.bambu-mobile.com/los-mejores-frameworks-para-desarrollo-de-apps-2022/>

[14] R. Isaac and B. Rivera, “UNIVERSIDAD TÉCNICA DE BABAHOYO ACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA CARRERA DE INGENIERÍA EN SISTEMAS DOCUMENTO PROBATORIO (DIMENSIÓN ESCRITA) DEL EXAMEN COMPLEXIVO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS,” 2021. <http://dspace.utb.edu.ec/bitstream/handle/49000/9473/E-UTB-FAFI-SIST-000153.pdf?sequence=1&isAllowed=y>

[15] P. sharma, “Las 10 principales ventajas de los servicios de desarrollo de Laravel para empresas,” Cynoteck, Dec. 13, 2019. [en línia] [consulta: 3 de gener de 2023]. Disponibilitat: <https://cynoteck.com/es/blog-post/top-10-advantages-of-laravel-development-services-for-enterprises/>

- [16] N. Bernat, “Flutter Pros and Cons 2022 - Summary and Recommendations,” leancode.co, Oct. 05, 2022. [en línia] [consulta: 3 de gener de 2023]. Disponibilitat: <https://leancode.co/blog/flutter-pros-and-cons-2022-summary>
- [17] AltexSoft, “The Good and the Bad of Flutter App Development,” AltexSoft, Ago. 04, 2022. [en línia] [consulta: 15 de desembre de 2022]. Disponibilitat: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-flutter-app-development/>
- [18] AltexSoft, “The Good and the Bad of Vue.js Framework Programming,” AltexSoft, Set. 11, 2019. [en línia] [consulta: 3 de gener de 2023]. Disponibilitat: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>
- [19] javatpoint, “Ionic Editors - javatpoint,” www.javatpoint.com. [en línia] [consulta: 24 de gener de 2023]. Disponibilitat: <https://www.javatpoint.com/ionic-editors>
- [20] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, “Agile Software Development: Methodologies and Trends,” International Journal of Interactive Mobile Technologies, vol. 14, no. 11, pp. 246–270, Jul. 2020, doi: 10.3991/ijim.v14i11.13269
- [21] M. Younas, D. N. A. Jawawi, I. Ghani, T. Fries, and R. Kazmi, “Agile development in the cloud computing environment: A systematic review,” Information and Software Technology, vol. 103, pp. 142–158, Nov. 2018, doi: 10.1016/j.infsof.2018.06.014
- [22] J. Izquierdo, “7 razones por las que debes utilizar metodologías ágiles,” Thinking for Innovation, Abr. 2022, [en línia] [consulta: 25 de gener de 2023]. Disponibilitat: <https://www.iebschool.com/blog/por-que-utilizar-metodologia-agiles-agile-scrum/>
- [22] A. Kavanaugh, “The Role of Progressive Overload in Sports Conditioning,” <http://teachfitnessconcepts.com/pdf%20files/the%20role%20of%20progressive%20overload%20in%20sports%20conditioning.pdf>
- [23] J. Romero, “Desarrollo de una aplicación android/móvil para la gestión y seguimiento de rutinas deportivas,” Universitat Politècnica de Valencia, 2021. <https://riunet.upv.es/bitstream/handle/10251/172232/Romero%20-%20Desarrollo%20de%20una%20aplicacion%20androidmovil%20para%20la%20gestion%20y%20seguimiento%20de%20rutinasdeport....pdf?sequence=1>

- [24] K. P. Zarzycki, “Diseño e implementación de una aplicación móvil para ayuda al entrenamiento deportivo,” Universitat Politècnica de Valencia, 2018. <https://riunet.upv.es/bitstream/handle/10251/111491/Zarzycki%20-%20Dise%C3%B1o%20e%20implementaci%C3%B3n%20de%20una%20aplicaci%C3%B3n%20m%C3%B3vil%20para%20ayuda%20al%20entrenamiento%20deportivo.pdf?sequence=1&isAllowed=y>
- [25] “What Is Node.js and Why You Should Use It,” Kinsta. [en línia] [consulta: 3 de gener de 2023]. Disponibilitat: <https://kinsta.com/knowledgebase/what-is-node-js/>
- [26] OpenWebinars.net. (2022). Qué es Visual Studio Code y qué ventajas ofrece. [en línia] [consulta: 4 de gener de 2023] Disponibilitat: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece>
- [27] EQengineered. (n.d.). 6 Considerations to Ensure Agile Team Value. [en línia] [consulta: 26 de gener de 2023] Disponibilitat: <https://www.eqengineered.com/insights/6-considerations-to-ensure-agile-team-value>
- [28] Sachdeva, Gurpreet. “Object Persistence with JPA | This Technology Life.” This Technology Life, 13 July 2020 [en línia] [consulta: 07 d’abril de 2023]. Disponibilitat: <https://thistechnologylife.com/object-persistence-with-jpa/>
- [29] “▷ Series Efectivas vs Series de Aproximación (GYM),” Mar. 02, 2023. [en línia] [consulta: 22 de maig de 2023]. Disponibilitat: <https://fitgeneration.es/series-efectivas-vs-series-de-aproximacio/>
- [30] S. Serna, Diseño de interfaces en aplicaciones móviles. Grupo Editorial RA-MA. https://books.google.es/books?hl=en&lr=&id=SI-fDwAAQBAJ&oi=fnd&pg=PP1&dq=dise%C3%B1o+aplicaciones+moviles&ots=bKh2ISp6ow&sig=I01PBIZUnE4pwKfRyYNcXP2OKZo&redir_esc=y#v=onepage&q=dise%C3%B1o%20aplicaciones%20moviles&f=false
- [31] J. A. Ortiz Zambrano, Y. T. Chavez Cujilan, M. del C. Toapanta Bernabe, and K. N. Lino Castillo, “La usabilidad y accesibilidad: Estudio de guías para aplicaciones en dispositivos móviles,” Dominio de las Ciencias, vol. 3, no. 3, pp. 1181–1209, 2017, <https://dialnet.unirioja.es/servlet/articulo?codigo=6244037>