



# UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO  
INGENIERÍA EN INFORMÁTICA.

## Desarrollo de un sistema para la organización de actividades para la práctica de lenguas extranjeras.

---

**Autor**  
Juan Salvador Molina Martín

**Director**  
Ignacio José Blanco Medina



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

—  
Granada, mes de 2021



# **Development of a system for the organization of activities for the practice of foreign languages.**

Juan Salvador Molina Martín

**Keywords:** mobile app, activity, competence, collaborative economy , Tuning, collaborative consumption, digital platform, Android Studio.

## **Abstract**

Nowadays, it is hard to think about a day without a mobile device. Due to the rising of those devices, more and more people use the many services that are available. On the other hand, the adaptation of the economics to the web environment is another main issue today.

Simoultaneously, learning new languages is becoming more necessary day by day either for work or leisure and due to globalization. More classical alternatives such as schools and languages academies fall behind when they face alternatives for learning languages, reaching teachers, places, competitive prices and so on.

The motivation for this project comes from the need for satisfying this growing demand from users who need to learn different languages without attending an academy and who look for another kind of opportunities. These kind of different services need to be available through a digital platform.

The main objective of this project is the development of an Android application allowing the organization of activities to be carried out in groups for the promotion of using foreign languages. In that way, the application will be able to configure activities and groups where users can participate.

The project will be developed in different stages. Firstly, the concepts of activities, skills and ideas on collaborative economics will be introduced. In order to represent those concepts, a data model will be proposed. Later on, the whole system for supporting the application will be developed and then implemented. Finally, details on the test of the application during the testing phase will be provided.

When using applications developed for mobile devices, visual aspect is a main issue. In the case of the proposed and developed application, lists for searching groups and for the activities that they organize make the user's experience to be easier. It is important to remark that searching for items by setting criteria and being able to get them ordered by different ways is essential for the user.

# **Desarrollo de un sistema para la organización de actividades para la práctica de lenguas extranjeras.**

Juan Salvador Molina Martín

**Palabras clave:** aplicación móvil, actividad, competencia, economía colaborativa, Tuning, consumo colaborativo, plataformas digitales, Android Studio.

## **Resumen**

Actualmente, resulta difícil un día a día sin Smartphone. Debido al auge de estos dispositivos, cada vez más personas utilizan los múltiples servicios que ofrecen. Por otro lado, la adaptación de la economía al entorno web, es otro de los aspectos más presentes en la actualidad.

Paralelamente, por efectos de la globalización, ya sea por trabajo u ocio, es cada vez más necesario el aprendizaje de nuevos, donde las opciones más tradicionales, escuelas y academias, se quedan atrás frente a alternativas en la que se puedan buscar idiomas, profesores, lugares cercanos, precios competitivos, etc.

La principal motivación de este proyecto viene dada por la necesidad de satisfacer esta demanda creciente en el usuario que quiere aprender cualquier idioma sin la necesidad de ir a una academia tradicional de idiomas a un coste relativamente bajo, quien puede proveer servicios para este tipo de actividades, que busca otro tipo de uso dada la situación actual. Todo esto soportado por una plataforma digital que los interconectan.

El presente proyecto tiene como objetivo general el desarrollo de una aplicación para Android que permite la organización de actividades en grupo que promueven la utilización de lenguas extranjeras. Por ello será capaz de crear actividades y grupos, en los que los otros usuarios pueden participar.

El desarrollo del proyecto contará con distintas fases, comenzando por un estudio de conceptos sobre actividades, competencias, economía y con un análisis para el diseño de un modelo de datos, continuando por un diseño del sistema y finalizando con la implementación de la misma, siguiendo los pasos marcados en las fases anteriores a fin de lograr los objetivos propuestos. Por último, una fase de prueba donde se ha detectado numerosos fallos que han sido corregidos.

La interfaz de usuario, por su componente visual, tiene una gran importancia en la aplicación, donde gran parte del protagonismo se centra en las listas detalladas de los grupos, sus actividades correspondientes, facilitando que busque el usuario. Además de la posibilidad de que el usuario pueda buscar una actividad o actividades concretas y ordenándolo a su función de una serie de criterios.

---

Yo, **Juan Salvador Molina Martín**, alumno de la titulación Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 75579353L, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Juan Salvador Molina Martín

Granada a X de mes de 2021 .



---

D. **Ignacio José Blanco Medina**, Profesor del Área de informática del Departamento ciencias de la computación e inteligencia artificial de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado *Desarrollo de un sistema para la organización de actividades para la práctica de lenguas extranjeras, Subtítulo del proyecto*, ha sido realizado bajo su supervisión por **Juan Salvador Molina Martín**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de mes de 2021 .

**El director:**

**Ignacio José Blanco Medina**

**DECLARACIÓN DE ORIGINALIDAD**

D./Dña. Juan Salvador Molina Martín

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2020-2021, es original, entendida esta, en el sentido de que no ha utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 5 de septiembre de 2021

Fdo: Juan Salvador Molina Martín

## **Agradecimientos**

En primer lugar, me gustaría agradecer a Ignacio José Blanco Medina por ofrecerse a ser tutor de este proyecto, que sin su ayuda este trabajo no hubiera sido posible. Además de tratarme siempre con mucha cercanía, intentando en todo tiempo ayudarme a resolver las dudas.

En segundo lugar, Agradecer a Rosana Montes Soldado la gran ayuda prestada en el inicio del proyecto, idea y diseño y experiencia de usuario durante su asignatura de diseño de interfaces de usuario.

Me gustaría agradecer también a mi compañero Ignacio Morillas Padial la gran ayuda prestada, para la concepción de la idea y desarrollo del diseño fue primordial.

Agradecer también al profesor Jesús García Miranda por el gran interés mostrado hacia el proyecto, su ayuda prestada en el desarrollo del proyecto y su inestimable apoyo, a través de una de las posibles soluciones, para la resolución de la matriz de actividades y competencias en el análisis y modelado.



# Índice general

Agradecimientos	IX
Índice de figuras	XV
Índice de tablas	XIX
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Planificación temporal . . . . .	2
1.4. Planificación de recursos . . . . .	3
1.4.1. Recursos humanos . . . . .	3
1.4.2. Recursos materiales . . . . .	4
1.5. Planificación presupuestaria . . . . .	4
<b>2. Fundamentos y bases tecnológicas</b>	<b>5</b>
2.1. Competencias y actividades . . . . .	5
2.1.1. Conceptos básicos . . . . .	5
2.1.2. Competencias genéricas. ¿Qué son?¿Qué tipos hay? . . . . .	7
2.1.3. Metodologías para el desarrollo de competencias genéricas . . . . .	9
2.1.4. El papel del contexto del aprendizaje en las competencias genéricas . .	11
2.1.5. Desarrollo en profundidad de las competencias genéricas en Dare+ .	12
2.1.6. Ejemplo de actividades para el desarrollo de competencias genéricas .	15
2.2. Economía Colaborativa . . . . .	20
2.2.1. Conceptos básicos . . . . .	20
2.2.2. Origen y concepto . . . . .	21
2.2.3. Pilares . . . . .	22
2.2.4. Tipos . . . . .	22
2.2.5. Modelo <i>peer to peer</i> . . . . .	23
2.2.6. Plataformas colaborativas . . . . .	24
<b>3. Análisis y modelado</b>	<b>27</b>
3.1. Análisis de actividades . . . . .	27
3.1.1. Selección inicial de actividades . . . . .	27
3.1.2. Análisis de las actividades recogidas . . . . .	31
3.2. Modelado de los parámetros y valores. . . . .	35
3.2.1. Parámetros y valores. . . . .	35
3.2.2. Tipos de actividades. . . . .	40
3.2.3. Competencias. . . . .	41
3.2.4. Configuración. . . . .	42
3.3. Modelado de las entidades <i>Usuario, Grupo y Lugar</i> . . . . .	46
3.3.1. <i>Usuario</i> . . . . .	46

## *Índice general*

3.3.2. <i>Grupo</i> . . . . .	46
3.3.3. <i>Lugar</i> . . . . .	48
3.4. Modelado de la actividad. . . . .	50
3.4.1. Actividad . . . . .	50
3.5. Modelado de las relaciones entre entidades <i>Usuario-grupo y Parámetro</i> . . . . .	54
3.6. Ejemplo de funcionamiento del sistema . . . . .	57
<b>4. Análisis y diseño del sistema</b>	<b>61</b>
4.1. Especificación de los requisitos . . . . .	61
4.1.1. Requerimientos funcionales . . . . .	61
4.1.2. Requerimientos no funcionales . . . . .	67
4.1.3. Restricciones . . . . .	67
4.2. Diseño de la arquitectura . . . . .	68
4.2.1. Esquema general de la arquitectura de la aplicación . . . . .	68
4.2.2. Diseño técnico de la arquitectura de la aplicación . . . . .	69
4.2.3. Capa de presentación . . . . .	71
4.2.4. Capa de aplicaciones . . . . .	75
4.2.5. Capa de datos . . . . .	79
4.3. Diseño de la seguridad . . . . .	81
4.4. Diseño de la interfaz . . . . .	81
4.4.1. Introducción . . . . .	81
4.4.2. Competitive Analysis . . . . .	82
4.4.3. Feedback Capture Grid . . . . .	82
4.4.4. Tasks and Sitemap . . . . .	83
4.4.5. Labelling . . . . .	85
4.4.6. Wireframes . . . . .	86
4.4.7. Logotipo . . . . .	87
4.4.8. Style Guide . . . . .	87
4.4.9. Guidelines . . . . .	88
4.4.10. Maqueta final. . . . .	89
<b>5. Diseño del sistema de recomendación</b>	<b>91</b>
5.1. Categorías . . . . .	91
5.2. Consulta simple o compuesta . . . . .	92
5.3. Criterio de búsqueda . . . . .	92
5.4. Incorporación de nuevos ranking . . . . .	93
<b>6. Implementación del sistema</b>	<b>95</b>
6.1. Bibliotecas de Android Studio usadas durante la implementación del sistema.	95
6.2. Construcción de la base de datos . . . . .	96
6.2.1. Tablas . . . . .	96
6.2.2. Views . . . . .	96
6.3. Interfaz . . . . .	97
6.4. Parte lógica . . . . .	98
<b>7. Fase de pruebas</b>	<b>101</b>
7.1. Pruebas en el cliente . . . . .	101
7.2. Pruebas en el servidor . . . . .	102

*Índice general*

<b>8. Conclusión y trabajos futuros</b>	<b>103</b>
8.1. Conclusiones . . . . .	103
8.2. Desviaciones temporales y presupuestarias . . . . .	104
8.3. Futuros proyectos . . . . .	105
<b>A. Primer apéndice</b>	<b>107</b>
A.1. Capa de datos del diseño de la arquitectura. . . . .	107
A.2. Manual de uso del jdbc postgres. . . . .	113
A.3. Wireframes:Bocetos de la interfaz. . . . .	115
A.4. Manual de usuario de la aplicación. . . . .	127
A.5. Manual de instalación de la aplicación. . . . .	132
<b>Bibliografía</b>	<b>133</b>



# Índice de figuras

1.1. Planificación de las fases. . . . .	3
2.1. Bases de las actividades . . . . .	6
2.2. Mapa gráfico de las bases de la metodología . . . . .	9
2.3. Gráfico del contexto de Smitter . . . . .	11
2.4. Características según Vázquez [JS98] . . . . .	12
2.5. Esquema sistemático de las bases de la competencia comunicativas. . . . .	13
2.6. Esquema de las bases del liderazgo. . . . .	13
2.7. Flexibilidad de las actividades propuestas según Smitter. . . . .	19
2.8. Esquema del consumo colaborativo. [Wah17] . . . . .	23
3.1. Matriz de actividades y competencias . . . . .	27
3.2. Matriz de actividades y competencias reordenada . . . . .	28
3.3. Matriz con al menos cuatro unos . . . . .	28
3.4. Matriz resultante . . . . .	28
3.5. Primer subconjunto del modelo . . . . .	29
3.6. Segundo subconjunto del modelo . . . . .	30
3.7. Configuración de dos de las sesiones. . . . .	31
3.8. Configuración de la actividad. . . . .	32
3.9. Configuración de la actividad. . . . .	32
3.10. Configuración de la actividad. . . . .	32
3.11. Configuración de la actividad. . . . .	33
3.12. Configuración de la actividad. . . . .	34
3.13. Entidad parámetro. . . . .	35
3.14. Entidad valor. . . . .	36
3.15. Entidad lógica, entero, etiqueta, real . . . . .	38
3.16. Entidad tipo de actividad. . . . .	40
3.17. Entidad competencia. . . . .	41
3.18. Entidad configuración. . . . .	42
3.19. Relación entre configuración y competencia . . . . .	43
3.20. Relación entre configuración y TipoActividad . . . . .	43
3.21. Relación entre configuración y Parámetro . . . . .	44
3.22. Relación entre configuración y Valor . . . . .	44
3.23. Entidad Usuario. . . . .	46
3.24. Entidad Grupo. . . . .	47
3.25. Relación Crea. . . . .	47
3.26. Relación forma. . . . .	48
3.27. Entidad Lugar. . . . .	48
3.28. Relación cede. . . . .	49
3.29. Entidad Actividad. . . . .	50
3.30. Relación Crea. . . . .	51
3.31. Relación Participa. . . . .	51

## *Índice de figuras*

3.32. Relación Se organiza para . . . . .	52
3.33. Relación Se realiza . . . . .	52
3.34. Relación Tiene . . . . .	53
3.35. Relación LugarTieneParámetro y GrupoTieneParámetro . . . . .	54
3.36. Relación LugarTieneValorParámetro y GrupoTieneValorParámetro . . . . .	55
3.37. Modelo de información completo . . . . .	56
4.1. Diagrama de casos de uso para un usuario no registrado . . . . .	62
4.2. Diagrama de casos de uso para un usuario registrado . . . . .	63
4.3. Diagrama de casos de uso para el administrador . . . . .	64
4.4. Esquema genérico para una arquitectura por niveles . . . . .	69
4.5. Ilustración simplificada del ciclo de vida de una actividad[dAe] . . . . .	70
4.6. Arquitectura de la aplicación web con las tecnologías empleadas . . . . .	71
4.7. directorio res . . . . .	72
4.8. Base de datos en postgresSql generado por DBeaver . . . . .	80
4.9. Tabla comparativa de empresas . . . . .	82
4.10. Malla receptora de información . . . . .	83
4.11. Jerarquía principal . . . . .	84
4.12. Vista principal de la aplicación . . . . .	86
4.13. Logotipo de la aplicación . . . . .	87
4.14. Ejemplo 1 . . . . .	88
4.15. Ejemplo 2 . . . . .	88
4.16. Inicio de sesión . . . . .	89
4.17. Registrarse . . . . .	89
5.1. Esquema de navegación . . . . .	92
5.2. Esquema de navegación . . . . .	93
5.3. Esquema de navegación . . . . .	93
5.4. Esquema de navegación . . . . .	93
6.1. Bibliotecas usadas durante la implementación . . . . .	95
6.2. Recursos para la inferior de navegación . . . . .	97
6.3. Recurso para la conexión con la actividad principal . . . . .	97
6.4. Recurso para la conexión con la con los fragmentos . . . . .	98
6.5. Cambio de actividad y envío de datos . . . . .	98
6.6. Recepción de la actividad e información . . . . .	99
6.7. Llamada a las funciones de una actividad desde un fragmento . . . . .	99
8.1. Planificación final . . . . .	104
A.1. Vista principal de la aplicación . . . . .	115
A.2. Vista del registro del usuario . . . . .	116
A.3. Vista del inicio de sesión . . . . .	117
A.4. Vista del perfil del usuario . . . . .	118
A.5. Vista de creación del grupo . . . . .	119
A.6. Vista de creación del establecimiento . . . . .	120
A.7. Vista de creación de la actividad . . . . .	121
A.8. Vista de la búsqueda . . . . .	122
A.9. Vista de los grupos . . . . .	123

*Índice de figuras*

A.10.Vista de los grupos pertenecientes . . . . .	124
A.11.Vista de las actividades. . . . .	125
A.12.Vista de las actividades pertenecientes . . . . .	126
A.13.Pantalla de inicio. . . . .	127
A.14.Pantalla de registro. . . . .	127
A.15.Pantalla de los grupos. . . . .	128
A.16.Pantalla de las actividades. . . . .	128
A.17.Pantalla de creación de un establecimiento. . . . .	129
A.18.Vista del perfil. . . . .	129
A.19.Pantalla de creación de los grupos. . . . .	130
A.20.Pantalla de creación de las actividades. . . . .	130
A.21.Pantalla de búsqueda de actividades. . . . .	131



# Índice de tablas

1.1. Costes iniciales . . . . .	4
3.1. Entidad parámetro . . . . .	36
3.2. Entidad Valor . . . . .	37
3.3. Entidad Etiqueta 3.15 . . . . .	39
3.4. Entidad lógica 3.15 . . . . .	39
3.5. Entidad Real 3.15 . . . . .	39
3.6. Entidad Entero 3.15 . . . . .	40
3.7. Entidad Usuario . . . . .	57
3.8. Entidad Lugar . . . . .	57
3.9. Relación cede . . . . .	57
3.10. Relación LugarTieneParámetro . . . . .	57
3.11. Relación LugarTieneValorDeParámetro . . . . .	57
3.12. Entidad Grupo . . . . .	58
3.13. Relación GrupoTieneParámetro . . . . .	58
3.14. Relación GrupoTieneValorParámetro . . . . .	58
3.15. Relación Crea . . . . .	58
3.16. Relación Forma . . . . .	58
3.17. Entidad Actividad . . . . .	58
3.18. Relación Crea . . . . .	58
3.19. Relación Participa . . . . .	59
3.20. Relación SeOrganizaPara . . . . .	59
3.21. Entidad Configuración . . . . .	59
3.22. Relación SeRealiza . . . . .	59
3.23. Relación LugarTieneValorDeParámetro . . . . .	59
3.24. Tiene . . . . .	59
3.25. Entidad Competencias . . . . .	59
3.26. Relación TieneCompetencias . . . . .	60
3.27. Entidad TipoActividad . . . . .	60
3.28. Relación Contiene . . . . .	60
3.29. Relación ConfiguraciónTieneParámetro . . . . .	60
3.30. Relación ConfiguraciónTieneValorParámetro . . . . .	60
4.1. Tabla caso de uso: Registrarse . . . . .	64
4.2. Tabla caso de uso: Introducir credenciales . . . . .	64
4.3. Tabla caso de uso: Ver perfil . . . . .	64
4.4. Tabla caso de uso: Cerrar sesión . . . . .	65
4.5. Tabla caso de uso: Establecer un lugar . . . . .	65
4.6. Tabla caso de uso: Buscar . . . . .	65
4.7. Crear un grupo . . . . .	65
4.8. Crear una actividad . . . . .	66
4.9. Tabla caso de uso: Formar parte de un grupo . . . . .	66

## *Índice de tablas*

4.10. Tabla caso de uso: Formar parte de una actividad.	66
4.11. Gestionar parámetros.	67
4.12. activitysign.xml	72
4.13. activitylogin.xml	72
4.14. activitybusqueda.xml	72
4.15. activitycrearactividad.xml	73
4.16. activitycrearestablecimiento.xml	73
4.17. activitycreargrupo2.xml	73
4.18. activitymain.xml	73
4.22. fragmentmygroup.xml	74
4.19. activityperfil.xml	74
4.20. fragmentactivity.xml	74
4.21. fragmentmyactivity.xml	74
4.23. fragmentgroup.xml	74
4.24. Listas dinámicas.	75
4.25. bottomnavmenu.xml	75
4.26. Data Class del sistema.	75
4.27. ActividadAdaptador.kt	76
4.28. Busqueda.kt	76
4.29. conect.kt	76
4.30. CrearActividad.kt	76
4.31. CrearActivityEstablecimiento.kt	77
4.32. CrearGrupoActivity2.kt	77
4.33. GrupoAdaptador.kt	77
4.34. Login.kt	77
4.35. MainActivity.kt	78
4.36. MiActividadAdaptador.kt	78
4.37. MiGrupoAdaptador.kt	78
4.38. PerfilActivity.kt	78
4.39. Sign.kt	78
4.40. ActivityFragment.kt	79
4.41. MyActivityFragment.kt	79
4.42. GroupFragment.kt	79
4.43. MyGroupFragment.kt	79
4.44. Información de la tabla Actividad.	81
4.45. Matriz De Tareas y Usuarios.	84
4.46. Labelling de la aplicación.	85
7.1. Tabla de pruebas de la aplicación.	101
7.2. Tabla de pruebas de la aplicación.	102
8.1. Costes reales.	105
A.1. Información de la tabla Competencias.	107
A.2. Información de la tabla Configuracion.	107
A.3. Información de la tabla Grupos.	107
A.4. Información de la tabla Lugar.	108
A.5. Información de la tabla Parámetro.	108

A.6. Información de la tabla TipoActividad. . . . .	108
A.7. Información de la tabla Usuario. . . . .	108
A.8. Información de la tabla ValorDebil . . . . .	109
A.9. Información de la tabla valorEntero. . . . .	109
A.10. Información de la tabla valorEtiqueta. . . . .	109
A.11. Información de la tabla valorLogico. . . . .	109
A.12. Información de la tabla valorReal. . . . .	110
A.13. Información de la tabla ConfiguracionTieneCompetencias. . . . .	110
A.14. Información de la tabla ConfiguracionTieneTipoActividad. . . . .	110
A.15. ConfiguracionTieneValorParametro. . . . .	111
A.16. Información de la tabla GrupoTieneParametro. . . . .	111
A.17. Información de la tabla GrupoTieneValorParametro. . . . .	111
A.18. Información de la tabla LugarTieneParametro. . . . .	111
A.19. Información de la tabla LugarTieneValorParámetro. . . . .	112
A.20. Información de la tabla actividadTieneConfiguración. . . . .	112
A.21. Información de la tabla configuraciontieneparametro . . . . .	112
A.22. Información de la tabla seRealiza. . . . .	113
A.23. Información de la tabla usuarioformaparteactividad. . . . .	113
A.24. Información de la tabla usuarioformapartedelgrupo. . . . .	113



# 1. Introducción

En este primer capítulo desarrollaremos la motivación por la cual se ha desarrollado este proyecto, los objetivos que vamos a cubrir y una planificación para cada uno de estos. Por otro lado, describiremos los recursos materiales y humanos que son necesarios para poder llevar a cabo este proyecto.

## 1.1. Motivación

El mundo en que vivimos es el efecto de la globalización, que entendemos como el concepto de la integración mundial en distintos ámbitos (económico, cultural, social, político y tecnológico) que se ha creado a mediados del siglo XX, haciendo un mundo más interconectado.

Una de las consecuencias inmediatas es el dominio de una serie de idiomas (Inglés, Chino, Español, Chino, Alemán), ya sea por su contexto histórico o por su fuerte economía, con las que se trabajan. Un ejemplo de todo esto es que hace 70 años un trabajador o incluso "empresario mediano" sólo necesitaba el idioma de su región para poder trabajar mientras que hoy en día casi cualquier trabajador, y no digamos ya empresarios de cualquier rango, necesita conocer dos o más idiomas, no solo para trabajar sino para comunicarse con personas de distintos países.

Las distintas soluciones que se han dado a lo largo de estos años por parte de los gobiernos ha sido el estudio de uno o dos idiomas en las escuelas que ha ido mejorando con el tiempo, iniciativas privadas como fue la creación de academias de idiomas para reforzar el aprendizaje o inclusión estudiar otros lenguajes. Paralelamente, surge una metodología de aprendizaje llamada "Conversación con nativos" o intercambio de idiomas, que son reuniones para practicar otros idiomas con gente del extranjero en bares o pubs.

A esto tenemos que sumarle que las personas tenemos una apretada agenda, teniendo que conciliar con el trabajo y la familia, que entra en conflicto con la educación dado que es una actividad que requiere tiempo, por ello buscan alternativas en la que puedan buscar idiomas, profesores, lugares cercanos, precios competitivos y alguna clasificación o ranking en función de los gustos de los usuarios. Si no encuentran todas estas características buscarán otra opción.

La principal motivación de este proyecto viene dada por la necesidad de satisfacer esta demanda creciente en el usuario que quiere aprender cualquier idioma sin la necesidad de ir a una academia tradicional de idiomas a un coste relativamente bajo, quien puede proveer servicios para este tipo de actividades, que busca otro tipo de uso dada la situación actual. Todo esto soportado por una plataforma digital que los interconectan.

Este tipo de prácticas, que hemos visto al principio, no estaban siendo aprovechadas o explotadas hasta que surge el concepto de "Economía colaborativa" en estos últimos años gracias a las plataformas digitales. A través de esta práctica, se desarrolla el concepto de "consumo colaborativo" que trata de intercambiar bienes o servicios entre particulares a cambio de una compensación no necesariamente monetaria pactada entre las partes. Un Ejemplo de ello, ha sido Airbnb y Uber. Podemos detectar que existen elementos comunes que permiten adaptar el concepto a los negocios de las academias de idiomas. Vemos una relación entre los

## *1. Introducción*

productores que corresponde al profesor o academia y consumidores que corresponde con los alumnos. En el ejemplo de intercambios de idiomas surge un tercer elemento ,aunque este se puede considerar otro producto, que corresponde con el establecimiento que se les "prestaba" para aprender el idioma. Si buscamos entre las principales competidores vemos que satisfacen algunas de estas demandas apoyándose en plataformas digitales pero en pocas de ellas se aprecia el verdadero potencial que tiene, dado que, con esta relación constante entre productores y consumidores, se ofrecen grandes ventajas. Estas se presentan no solo para el usuario sino también para el profesorado a la hora de dinamizar su enseñanza y buscar la mayor facilidad para los integrantes a través de las plataformas.

## **1.2. Objetivos**

El presente proyecto tiene como objetivo general el desarrollo de una aplicación para Android que permite la organización de actividades en grupo que promueven la utilización de lenguas extranjeras. La plataforma permitirá la organización de actividades, propondrá la configuración del grupo de asistentes y pondrá en contacto a los diferentes roles (usuarios, profesionales y establecimientos). Se seguirá una metodología de diseño de la solución y desarrollo de software escogida entre el alumno y el tutor.

Los objetivos específicos del proyecto son:

1. Estudiar diferentes tipos de actividades para la promoción del uso de las lenguas extranjeras, así como su configuración.
2. Definir un modelo para las diferentes entidades que van a interactuar en el sistema.
3. Diseñar una arquitectura para el sistema propuesto.
4. Evaluar algoritmos para que el sistema proponga la configuración de actividad y grupo más adecuada para cada fin.
5. Implementar los diferentes elementos de la arquitectura diseñada.
6. Diseñar e implementar una interfaz enfocada en las necesidades del usuario.

## **1.3. Planificación temporal**

La planificación inicial del proyecto estará estructurada en un total de siete fases, las cuales muestran el desarrollo y metodología usada durante el proyecto.

La primera fase comprende al estudio del problema, los objetivos establecidos y su planificación. Durante esta fase las prioridades fueron encontrar un marco en el que el proyecto pudiera asentarse y planificar a largo plazo cómo iba a ser desarrollado. Esta fase llevará un total de tres semanas. Cada semana se destinarán 3 horas.

La segunda fase se centrará en el estudio de distintos escenarios (actividades) para la promoción de las lenguas extranjeras, además de las diferentes metodologías de aprendizaje y los fundamentos de la economía colaborativa. Esta fase llevará un total de cuatro semanas\*. Cada semana se destinarán 10 horas.

La tercera fase llevaremos a cabo el análisis y modelado de los roles que aparecen en los distintos escenarios, y la comunicación entre ellos. Esta fase llevará un total de cuatro semanas. Cada semana se destinarán 12 horas.

La cuarta fase corresponde con el diseño de la arquitectura del sistema y de la interfaz de usuario. Esta fase se dividirá en dos fases y cada una 4 semanas cada una.

La quinta fase está enfocada en el diseño e implementación de un sistema de recomendación para la configuración de actividades. Esta fase llevará un total de dos semanas.

La sexta fase comenzará con la implementación de los elementos diseñados en la arquitectura y para la interfaz de usuario. Esta fase se dividirá en dos sub-fases de 4 semanas cada una.

Por último, la fase de prueba tendrá una duración estimada de dos semanas, en las que se probarán y analizarán las funcionalidades diseñadas e implementadas. Cada semana se destinarán 10 horas.

La documentación del proyecto se desarrollará a lo largo de todo el proyecto. Se destinarán 2 horas semanales al menos para poder llevarlo a cabo y corregir errores.

En resumen, cada semana tendrá una dedicación de 14 horas semanales, aunque también puede verse modificado por la fase en la que nos encontramos o por falta de tiempo.

En la figura 1.1, podemos ver un diagrama de las diferentes fases del proyecto.

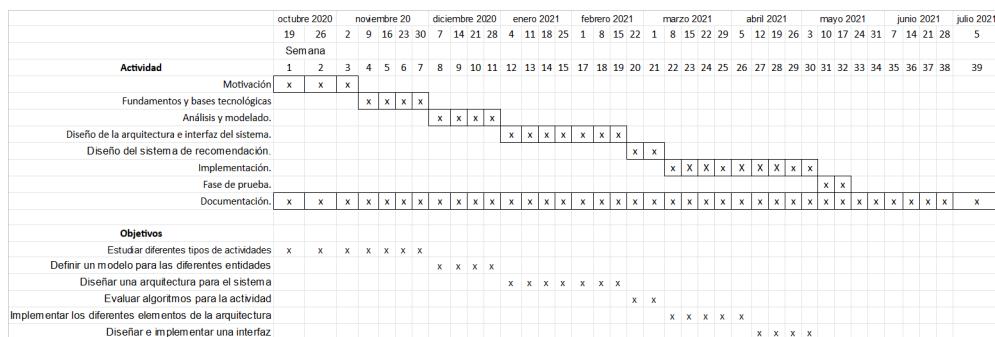


Figura 1.1.: Planificación de las fases.

## 1.4. Planificación de recursos

### 1.4.1. Recursos humanos

Atendiendo a la planificación temporal, tenemos un total de 41 semanas repartidas a lo largo de diez o nueve meses para el diseño y el desarrollo del prototipo desde su fase inicial hasta su fase más avanzada para probarla. Teniendo en cuenta, las semanas de las que disponemos y horas al día (2h \* 7 días/sem) tendremos un total de 14 horas semanales. Teniendo en cuenta todo esto, dispondremos de 600 horas para terminar el proyecto. El número de días y horas dedicadas al proyecto cambiarán, dependiendo de la fase en la que se encontraba el proyecto.

En la elaboración de este proyecto han participado:

- D. Juan Salvador Molina Martín, alumno de la Escuela Técnica Superior de Ingeniería Informática y Telecomunicaciones de la Universidad de Granada, como autor del proyecto.

## 1. Introducción

- D. Ignacio José Blanco Medina, profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, en calidad de tutor del proyecto.

### 1.4.2. Recursos materiales

Para la correcta elaboración del proyecto ha sido necesario disponer de un ordenador para poder trabajar y un lugar de trabajo.

- Ordenador personal: Asus Rog Strix G531GT-BQ012 Intel Core i7-9750H/16GB/512GB SSD/GTX1650/15.6"junto a periféricos: ratón, teclado, pantalla...
- Sistema operativo: windows 10.
- Herramientas trabajo: Licencia GSuite (Drive, Docs, Gmail, Meet...), Adobe XD, Android Studio.
- Espacio de Trabajo: Vivienda actual.
- Consumos de zona de trabajo: Energético e internet.

## 1.5. Planificación presupuestaria

En total se presupuestan 600 horas aproximadamente para la finalización del proyecto. Según el informe *indeed*[ind2o] sobre el salario medio de un informático junior asciende a cerca de 19.188€ al año. Sobre ese salario la empresa tiene que pagar la cuota patronal de seguridad social un 23 '60. Por tanto, los costes laborales anuales de un informático junior son 23.713€ al año. En el Estatuto de los Trabajadores, establece una jornada máxima de 1.828 horas anuales. Por lo tanto ,el sueldo medio por hora sería de 12,97 ≈ 13 €/h.

Adicionalmente, a lo largo de la elaboración y ejecución del proyecto se ha estimado un total de 40 tutorías de 1 hora de duración aproximadamente. Considerando los honorarios del Dr. Ingeniero Informático como tutor de este proyecto, estos ascienden a 40 €/h.

El detalle de lo expuesto anteriormente, así como el cálculo total, se recogen en la tabla 1.1.

Concepto	Coste	Tipo de pago	Duración	Total
<b>Recursos Materiales:</b>				
Asus Rog Strix	899,00€	Unico		899,00€
Sistema operativo	145,00€	Unico		145,00€
Herramientas trabajo	0€	Unico		0€
Espacio de Trabajo	80,00€	Mensual	10	800€
<b>Recursos Humanos:</b>				
Trabajo del proyecto	13€	Mensual	600	7800€
Tutorias del profesor	40€	Mensual	80	3200€
<b>Total(€)</b>				<b>12844€</b>

Tabla 1.1.: Costes iniciales.

## 2. Fundamentos y bases tecnológicas

En este segundo capítulo nos centraremos en el estudio de las competencias y actividades, y por otro lado la economía colaborativa.

### 2.1. Competencias y actividades

Las actividades que realizamos las personas para nuestro desarrollo, ya sea la finalidad que buscamos, están sustentadas en unas competencias, una metodología y en un tipo de aprendizaje.

En nuestro caso, las actividades que nos interesa están relacionadas con el aprendizaje de cualquier lengua extranjera. Por lo tanto, nos centraremos en llevar a cabo el desarrollo de una serie de competencias genéricas, aplicar un tipo de metodología de enseñanza, un aprendizaje informal dado que la naturaleza de nuestro proyecto está fuera de las competencia de las instituciones educativas y como novedad, incorporar la economía colaborativa para establecer una relación entre iguales en las actividades que se desarrollarían.

Nuestro objetivo aquí, es relacionar las actividades con las tres bases anteriores con el fin de obtener una síntesis de una serie de actividades que nos ayude en nuestro problema presentando, e incorporar un nuevo tipo de economía, lejos de la tradicional, en la que encontraremos bastantes similitudes con el punto de convertirla en otro pilar de las actividades.

Para todo esto, nos apoyaremos en estudios e iniciativas como *Tuning* y *Dare+*, comprendemos diferentes metodologías que se pueden aplicar y entenderemos los conceptos básicos de economía colaborativa.

#### 2.1.1. Conceptos básicos

El objetivo de este apartado nos ayudará a entender algunos conceptos que se desarrollarán a lo largo de este capítulo.

**Actividad [act]** Conjunto de tareas o acciones realizadas por un ser vivo, que las desarrolla impulsado por el instinto, la razón, la emoción, o la voluntad, hacia un objetivo. En este contexto, hablaríamos de actividades que nos ayuden al desarrollo de unas competencias más elevadas dejando atrás cualquier actividad básica.

**Tuning [JG]** Es una iniciativa de las universidades para las universidades. Se busca iniciar un diálogo para intercambiar información y para mejorar la colaboración entre las instituciones de educación superior, favoreciendo el desarrollo de la calidad, de la efectividad y de la transparencia.

Dare+

**Cognoscitiva [Rev]** En este contexto hablamos de un modelo explicativo sobre el aprendizaje basado en el concepto de la “acción”, de la experiencia.

## 2. Fundamentos y bases tecnológicas



Figura 2.1.: Bases de las actividades

**Metodológica [met]** Son estrategias que se ponen en marcha para conseguir alcanzar de forma adecuada los objetivos y contenidos previstos.

**Habilidad [HAB]** Son posibilidades que tiene la persona para realizar una acción, las haya llevado a cabo o no, es como el potencial, es reunir las cualidades para hacer algo.

**Destreza [Raf]** Es la capacidad con que una persona lleva a cabo de manera satisfactoria una tarea o un trabajo.

**Aprendizaje formal [Raf]** Se refiere al aprendizaje que se lleva a cabo dentro de un proceso institucionalizado, secuencial y estandarizado, llevado a cabo desde las escuelas, institutos o universidades, atendiendo a unos objetivos didácticos, con una duración determinada y que concluye con un proceso de evaluación y certificación.

**Aprendizaje no formal [Raf]** Es el que se lleva a cabo en cualquier actividad educativa organizada y sistemática realizada fuera del sistema educativo reglado, y sin que haya una voluntad certificadora expresa. No obstante, tiene carácter estructurado.

**Aprendizaje informal [Raf]** Es aquel que se adquiere en las actividades cotidianas relacionadas con el trabajo, la familia o el ocio. No es un aprendizaje estructurado (carece de objetivos didácticos, no tiene una duración determinada ni un soporte concreto), no se evalúa y, evidentemente, no conlleva ninguna certificación.

**Contextos de Aprendizaje [Raf]** Hace referencia a los tres principales contextos: formal, no formal e informal.

**Aprendizaje [Raf]** Se entiende por aprendizaje al proceso a través del cual el ser humano adquiere o modifica sus habilidades, destrezas, conocimientos o conductas, como fruto de la experiencia directa, el estudio, la observación, el razonamiento o la instrucción.

### 2.1.2. Competencias genéricas. ¿Qué son? ¿Qué tipos hay?

En la actualidad las empresas, las organizaciones y la sociedad en su conjunto demanda a los trabajadores o profesionales el manejo de determinadas competencias, habilidades y destrezas que hacen que la actividad profesional sirva para resolver diferentes problemas y anticiparse a ellos ya sea en el campo de la economía, salud o sociedad [yJlo8].

Pero, ¿Qué entendemos por competencia?

Según el pedagogo alemán Bunk se define como la “capacitación real para resolver determinados problemas”, contando además de los conocimientos, destrezas y aptitudes, o flexibilidad autonomía, con las competencias“ humanas y sociopolíticas”, que “amplían el radio de acción a la participación en el entorno profesional, así como a la organización del trabajo y a las actividades de planificación [SER99]”.

Por otro lado, en el proyecto *Tuning* lo define como una combinación dinámica de conocimientos, comprensión, habilidades y destrezas [yPB17].

Las competencias abarcan desde niveles muy bajos, como puede ser lo imprescindibles para poder trabajar, a niveles más altos, como la es la especialización del trabajo.

Estas podemos agruparlas en cuatro categorías:

1. **Básicas.** Son aquellas competencias comunes en todos los trabajos y que resultan imprescindibles para incorporarse al mercado laboral.  
Estos hacen referencia al lenguaje, comunicación, matemáticas, ciencia, cultura. También entran en juego el análisis critico, la creatividad, tomar decisiones,etc..
2. **Técnico-profesionales.** Son aquellas competencias específicas de una profesión, que vienen facilitadas por la formación profesional del nivel correspondiente. Se resumiría en el dominio experto de la profesión.
3. **Claves.** Son aquellas competencias profesionales que resultan esenciales para formar parte activa y efectivamente en las nuevas formas de organización del trabajo que resultan de los recientes y profundos cambios derivados de los avances tecnológicos y de los cambios en el comportamiento de los consumidores.
4. **Transversales o genéricas.** Son aquellas que se consideran relevantes para todos los graduados de educación superior, independientemente de las especializaciones que hayan elegido, según Tuning [yPB17]. Esto quiere decir que no se encuentran ligadas a ningún tipo de trabajo específico sino que son necesarias e importantes en cualquier trabajo. Por eso, también se le puede llamar genéricas.

Esta ultima que hemos mencionado, podríamos considerarla la de importancia porque no importa el tipo de trabajo o la actividad que vayamos a resolver. Aquí se centra más en como las personas actuamos ante un problema, la capacidad para buscar una solución y el modo de resolverlo. Podemos entenderlo viendo como un cirujano y un operador de una central nuclear, sus competencias profesionales son distintas pero sus competencias transversales a la hora de tomar decisiones y resolver problemas en situaciones comprometidas son iguales.

## *2. Fundamentos y bases tecnológicas*

Podemos encontrar diferentes modelos a la hora de clasificarlos, pero nos centraremos en el modelo establecido en *Proyecto Erasmus Tuning* y *Proyecto Erasmus+ Dare+*, donde se divide en tres grupos diferentes.

- Instrumentales: Nos encontramos las herramientas para el desarrollo eficaz de la profesión. Dentro de aquí, vuelve a diferenciarlas en cognoscitivas, metodológicas, tecnológicas y lingüísticas.
  1. Capacidad de análisis y síntesis.
  2. Capacidad de organización y planificación.
  3. Conocimiento general básico.
  4. Profundización en el conocimiento básico de la profesión.
  5. Comunicación oral y escrito en el idioma propio.
  6. Conocimiento de una o más lenguas extranjeras.
  7. Habilidades básicas informáticas.
  8. Conocimientos de informática relativos al ámbito de estudio.
  9. Habilidades de gestión de la información (capacidad para recuperar y analizar información de diversas fuentes)
  10. Resolución de problemas
  11. Toma de decisiones
- Interpersonales: Se refieren a la interacción social y cooperación del titulado con su ámbito social: capacidad de exteriorizar los propios sentimientos, habilidad crítica y autocritica.
  1. Trabajo en equipo.
  2. Trabajo en un equipo de carácter interdisciplinar.
  3. Capacidad para trabajar en un contexto internacional.
  4. Capacidad para comunicarse con expertos de otros campos.
  5. Habilidades en las relaciones interpersonales.
  6. Razonamiento crítico.
  7. Compromiso ético.
  8. Capacidad de crítica y auto-critica.
  9. Apreciación de la diversidad y multiculturalidad.
- Sistémicas: Capacidades o habilidades de visión y análisis de realidades totales y multidimensionales:
  1. Capacidad para aplicar el conocimiento en la práctica.
  2. Capacidad de aprendizaje (adquirir experiencia).
  3. Aprendizaje autónomo.
  4. Adaptación a nuevas situaciones.
  5. Liderazgo.

## 2.1. Competencias y actividades

6. Conocimiento de otras culturas y costumbres.
7. Iniciativa y espíritu emprendedor.
8. Motivación por la calidad.
9. Sensibilidad hacia temas medioambientales.
10. Habilidades de investigación.
11. Capacidad para generar nuevas ideas (creatividad).
12. Diseño y gestión de proyectos.

Esta clasificación podemos verla con otro enfoque en función de la situación geográfica, ámbito de estudio y áreas de especialización [Tunoo].

En este caso, es interesante estudiar las metodologías aplicadas al conocimiento de lenguajes extranjeros pero, a la persona también le interesa adquirir otras competencias. Por ello, damos esta clasificación.

### 2.1.3. Metodologías para el desarrollo de competencias genéricas

Las metodologías pueden considerarse como el pilar o la base más importante de las actividades, dado que tenemos que establecer unas pautas o unos procedimientos para lograr el objetivo de nuestra práctica.

Puede que nos de la sensación de que seguir una metodología esté ampliamente relacionado con el aprendizaje formal o con unas estructuras institucionalizadas y estandarizadas. Pero estos, se pueden aplicar a cualquier actividad ya sea formal o informal.

En función del objetivo que estamos buscando, seamos conscientes o no de ello, aplicamos diferentes metodologías para poder cumplir el objetivo perseguido. Todo esto lo conseguimos a través de los métodos, técnicas y estrategias [BF17a].

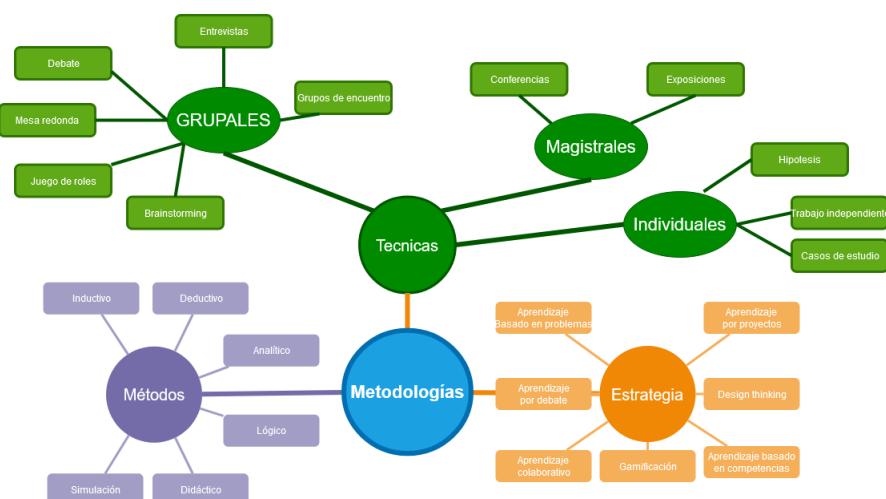


Figura 2.2.: Mapa gráfico de las bases de la metodología

Como hemos mencionado, cuando se ha estudiado diferentes actividades del proyecto *Erasmus Tuning* y *Erasmus+ Dare+* existen relaciones en la metodología aplicada. En el mapa gráfico dado, podemos observar que a la hora de querer construir una metodología,

## *2. Fundamentos y bases tecnológicas*

debemos escoger en función de las competencias genéricas que deseamos y que no entremos en una situación de incompatibilidad entre las técnicas y la estrategias.

Por ello estudiaremos y entenderemos las diferentes estrategias que hemos visto en la imagen y cual es su objetivo o al menos acercarnos a el.

- Aprendizaje basado en problemas. El líder del equipo expone un problema práctico o con una situación real y tienen que analizar y resolver el problema planteado de modo que se logren ciertos objetivos de aprendizaje [ABP].
- Aprendizaje por proyectos. El objetivo que se persigue es que un grupo, habitualmente de personas o solo uno, analice, desarrolle y solucione un problema real a través de un proyecto. El objetivo que se persigue es que se colabore en grupo y no solo se memorice, sino que aprenda con el proyecto [Ahh].
- Design thinking. Es un método para generar ideas innovadoras que centra en entender y dar solución las necesidades de los usuarios. En la educación, con esta técnica, busca identificar los problemas de cada componente del grupo y generar una mejor experiencia educativa [emp16].
- Aprendizaje colaborativo. El objetivo de esta práctica es común para todos los integrantes. Por tanto, cada uno desempeñará un rol en el grupo y se tendrá éxito si cada uno cumple con su trabajo [Hie].
- Aprendizaje basado en competencias. Se basa en que cada vez que aprendemos algo obtenemos siempre conocimiento, habilidades o destrezas. La estrategia que sigue es que se establece unos competencias que queremos aprender y no se pasan a otros hasta que estos no se han aprendido [com16].
- Gamificación. En esta estrategia, se integra las mecánicas de un videojuego en el aspecto de cumplir una serie de objetivos, para poder cumplir unos objetivos propuestos y poder así. Todo esto, para poder mejorar nuestro conocimiento o mejorar alguna habilidad [BF17b].
- Aprendizaje por debate. El objetivo principal es fomentar el discurso razonado. Se busca un tema de debate donde, se hace una búsqueda entre los diferentes equipos rivales y debatir sobre ello. Por último, también se busca que los integrantes intenten respetar los turnos de palabras e intervenir cuando sea necesario [rub].

La mayoría de estas estrategias siguen técnicas grupales, dado que son las que se tienen que hacer en grupo en la mayoría de los casos. Se busca que cada componente del grupo se relacione con el resto y sea un miembro activo, para que sea capaz de desarrollar una serie de competencias genéricas, además del objetivo buscado.

Suelen seguirse el mismo patrón para la mayoría de las estrategias o al menos tienen algunas en común:

- Presentación del problema.
- Información en común de los participantes.
- Análisis y desarrollo del problema.
- Búsqueda de la información necesaria.

- Debate de las posibles soluciones del problema.
- Presentación de la solución.

#### 2.1.4. El papel del contexto del aprendizaje en las competencias genéricas

Como ultimo pilar de las actividades, veremos que juega un papel fundamental en los proyectos *Erasmus Tuning* y *Erasmus+ Dare+* el contexto del aprendizaje. Para ello, entenderemos los diferentes tipos de contexto de aprendizaje que hay [Mar14] .

Estos contextos de aprendizaje se conocen como formal, no formal e informal, pero estos se distinguen en función de dos criterios, el primer criterio se basa en la organización de la educación en una secuencia de grados y niveles oficialmente reconocidos, mientras que el segundo criterio es en función de la programación de las acciones educativas según Smitter [Smio6].

El entiende que las diferencias que existen no son conceptos opuestos, sino como una linea continua donde se han colocado en función de la libertad y flexibilidad que da cada uno.

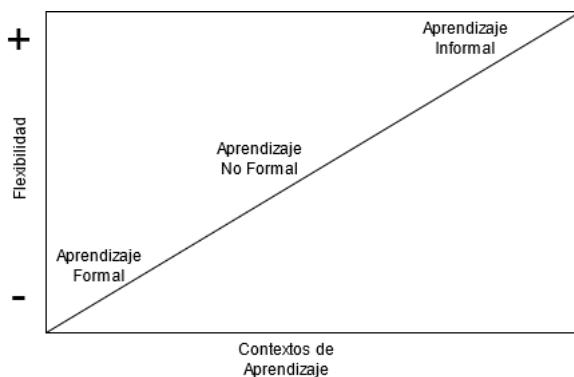


Figura 2.3.: Gráfico del contexto de Smitter

Según estos criterios estos se dividen en formal y no formal, y por otra parte, la informal.

La primera muestra una mayor organización y sistematización. A su vez estas tienen otros dos criterios para poder diferenciarse.

- Según el criterio estructural, el contexto formal está dentro de un sistema académico reglado por el estado, mientras el contexto no formal, pertenecen a actividades extraescolares ,cursos y talleres.
- Según el criterio metodológico, el contexto formal presenta una estructura escolar como la las clases presenciales, sistema de distribución y espacio propio, mientras el no formal se aleja de estos patrones.

Por otra parte, la segunda la considera como un proceso que se hace a lo largo de la vida, donde se adquiere conocimientos, habilidades y actitudes, mediante actividades cotidianas por lo tanto, no tiene una duración determinada ni un soporte concreto.

Existe otra forma de diferenciar su contexto en función de una serie de características según Vázquez [JS98] .

Con el proyecto *Erasmus+ Dare+* y *Erasmus+ Tuning* aprovecha toda la flexibilidad del contexto informal dado que todas las actividades que se presentan se realizan fuera de las

## 2. Fundamentos y bases tecnológicas

		Contexto de aprendizaje		
	Formal	No formal	Informal	
Estructuración	Instituciones Académicas	Programas y cursos	No tiene estructura	
universalidad	Incluye a todas las personas	En función de unas características comunes de un grupo	En función del nivel académico	
duración	Límite establecido por las instituciones académicas	En función de la duración del curso	Toda la vida del sujeto	
institución	Escuelas y universidades	Empresas y organizaciones	Instituciones débiles o inexistentes	

Figura 2.4.: Características según Vázquez [JS98]

instituciones universitarias usando uso de una estructura y objetivos definidos, ya que son proyectos de gran dimensión.

### 2.1.5. Desarrollo en profundidad de las competencias genéricas en Dare+

En el proyecto *Erasmus+ Dare+* que estamos estudiando para construir nuestro proyecto, se centra exclusivamente en siete competencias genéricas o transversales.

Se resumen en estas siete:

1. **Comunicación.** Se define como la capacidad y habilidad de una buena escritura y oración. No se limita simplemente a saber comunicarse y escribir sino tener la capacidad de transmitir una información ya sea una persona experta o que sea nuevo en eso.

Se considera la base más importante para el buen liderazgo, gestión y resolución de conflictos o problemas. Esta tiene como base cuatro competencias básicas :

Tendrá el código CO1.

2. **Trabajo en equipo.** Es la capacidad de integrar y colaborar de forma activa para obtener unos objetivos comunes con otras personas, áreas y organizaciones.

Para ello, requiere una serie de criterios:

- Compromiso con el objetivo compartido e identificado.
- Interdependencia y compromiso con el proceso de aprendizaje interpersonal y en equipo.
- Aportar opinión personal para entender y comprender su punto de vista.
- Conocimiento de cualidades personales e interpersonales, es decir, identificar fortalezas y debilidades propias para trabajar en equipo.
- Contextualizar la tarea asignada, es decir, explicarla tarea asignada.

Tendrá el código CO2

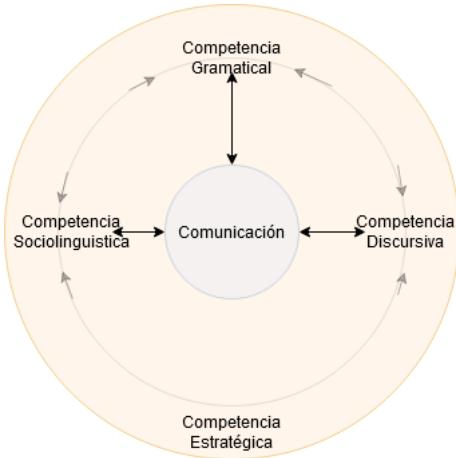


Figura 2.5.: Esquema sistemático de las bases de la competencia comunicativa.

3. **Liderazgo.** Es la capacidad para influir y guiar las acciones de un grupo hacia la visión en común y compartida, obteniendo el apoyo y compromiso del grupo con el objetivo de llegar a una meta significativa. Así mismo, tiene que dar ejemplo a través de su cumplimiento, ética y compromiso [SR17].

Para ello se suelen seguir los siguientes pasos:

- Tomar iniciativas y comunicarlas con convicción e integridad, estimulando a los demás.
- Trasmitir confianza y llevar a otros a la acción.
- Ejercer influencia en su propio entorno para lograr los objetivos deseados

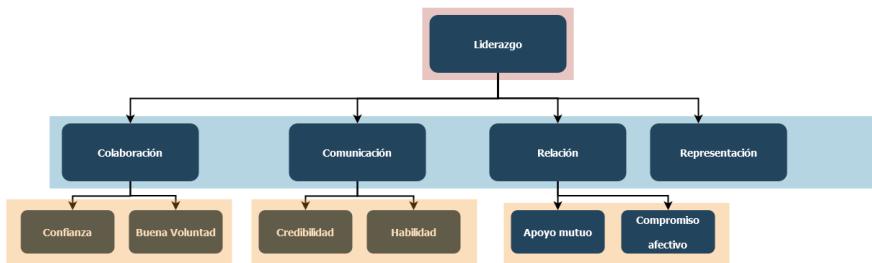


Figura 2.6.: Esquema de las bases del liderazgo.

Tendrá el código CO3

4. **Capacidad de apreciación de la diversidad y multiculturalidad.** Se define como la combinación dinámica de valores, actitudes, conocimiento, habilidad que nos permite comunicarnos de forma efectiva y apropiada con otras culturas e idiomas. Es una cualidad innata del ser humano la curiosidad por otras culturas y por construir un puente para conectar con otras culturas. Para que esto sea más efectivo esta conexión o choque entre culturas deben respetarse estos rasgos:

- Curiosidad por aprender de otras culturas.

## *2. Fundamentos y bases tecnológicas*

- Respeto entre las dos culturas y todos sus aspectos.
- Tener en cuenta las diferentes perspectivas.
- Empatía para poder comprender emociones y sentimientos.

Estos atributos se diferencian en tres niveles empezando con una aproximación desde lejos con el acercamiento entre dos culturas hasta adentrarse en lo más profundo de ellas. Estos se desarrollan en el proyecto *Dare+* en forma de rubricas [Yar17] .

Tendrá el código CO4

5. **Resolución de problemas y conflictos.** Es la capacidad de desarrollar y utilizar habilidades cognitivas, emocionales y de comportamiento con el fin de mejorar los resultados productivos del conflicto al tiempo que reducen la probabilidad de una escalada o daño. Los resultados de la competencia en conflictos incluyen una mejor calidad de las relaciones, soluciones creativas y acuerdos duraderos para abordar los desafíos y las oportunidades en el futuro.

Para abordar un conflicto se suelen dar estas aptitudes:

- Competitividad.
- Sumisión.
- Escapar.
- Cooperación.
- Compromiso.

Todo esto depende de como lo afrontemos, y si el objetivo es importante para nosotros, por ello de la idea de no afrontar los problemas. Además, en función de la relación con el resto también dependerá en actuar demostrando compromiso o no.

Tendrá el código CO5.

6. **Desarrollo y gestión de proyectos.** Es la capacidad para identificar una necesidad, analizar y definir el contexto y los recursos, y para diseñar, desarrollar, implementar y mejorar las ideas de acción con el objetivo de producir una respuesta apropiada y factible.

Indicadores que nos dicen que seguimos un buen camino en el desarrollo y gestión de un proyecto son los siguientes:

- Análisis detallado encontrando las necesidades y causas.
- Soluciones creativas e innovadoras.
- Llevar una buena estructura del proyecto y un portafolios.
- Definir un tiempo máximo para cada tarea asignada del proyecto.
- Control del presupuesto del proyecto.
- Comunicación con el trabajo en grupo.
- Monitorización de cada proceso del proyecto.

Tendrá el código CO6

7. **Emprendimiento social.** Es la capacidad que tenemos para dar soluciones innovadoras a un problema social. Se considera como una competencia genérica que engloba tres conceptos, la organización, la empresa y el liderazgo. Esto significa que para alcanzar y dominar esta competencia, debemos dominar otras competencias más básicas:

- Liderazgo.
- Optimismo.
- Carácter.
- Creatividad e innovación.
- Empatía.
- Inteligencia emocional
- Capacidad de adaptarse a cualquier obstáculo.

Tendrá el código CO7

#### 2.1.6. Ejemplo de actividades para el desarrollo de competencias genéricas

En este capítulo, nos centraremos en describir cada capítulo del libro *Developing Generic Competences Outside the University Classroom*.

Con los fundamentos adquiridos anteriormente, podremos dar una visión global de cada actividad que se presenta en el libro y poner en práctica las principales bases en las que se sustentan las actividades.

Dado la naturaleza del proyecto *Erasmus+ DARE+*, el contexto del aprendizaje de estas actividades se consideran no formales e informales.

A cada una de las nueve actividades que vamos a presentar se le asignará un código para más adelante.

Por ello daremos la descripción de nueve experiencias de alumnos de diferentes universidades y países que participaron en el proyecto *Erasmus+ DARE+*:

1. **Entrepreneurs To Change The World**[Fra17]. Esta actividad está relacionada con un proyecto de emprendimiento social. Este taller tiene el objetivo de integrar en los alumnos las competencias como la comunicación, liderazgo, creatividad, trabajo en equipo, desarrollo y gestión de proyectos.

Este taller consiste en 8 alumnos dividido en 4 sesiones con un total de 16 horas de actividades:

- La primera sesión consiste en enseñar a los alumnos que es el emprendimiento social y sus competencias a través de vídeos de Youtube. Seguido de actividades y test de personalidad en referencia a esto para solidificar las bases del alumno. Por último, los alumnos debatirán sobre las actividades presentadas.
- En la segunda sesión, los alumnos tienen que presentar un power point, donde explique las principales características de una persona creativa y como estos llegan a ideas ingeniosas. Se les pondrá vídeos para motivarlos. Para poner todo esto en práctica, tendrán actividades, como el Brainstorming.
- En la tercera sesión, los alumnos tendrán el objetivo de aprender como hacer un proyecto de emprendimiento social, con ayuda de modelos. A partir de aquí, empezarán a construir su propio proyecto.

## 2. Fundamentos y bases tecnológicas

- En la cuarta y última sesión, los alumnos expondrán su proyecto y responder a las preguntas del resto del alumnado. Por último, tendrán un debate sobre lo aprendido en la actividad.

En esta actividad hemos podido observar la siguiente metodología:

- Los alumnos podían organizarse en grupos o actuar en solitario.
- Se han aplicado la estrategia de aprendizaje basado en competencias dado que ese era el principal objetivo.
- El método seguido es el deductivo dado que han partido de concepto generales hasta construir su propio proyecto personal.

Tendrá el código AC1.

## 2. Learning Through An Educational Simulator[BF17c]

Esta actividad llamada *vLeader* es un programa virtual en el que los participantes pueden asumir el rol de ser un jefe de negocios e interactuar con bots inteligentes. Esta simulación se compone de varios niveles con el objetivo hacer que el usuario interaccione para que sea evaluado.

El usuario o jugador tiene una serie de objetivos que tiene que cumplir, como si fuera un videojuego, y cumplirlos participando y liderando en la simulación.

Esta actividad consisten en 36 alumnos en 7 grupos de 4 a 6 alumnos. Esta se divide en dos partes:

- La primera parte consiste en la adaptación del participante a la simulación cumpliendo una serie de objetivos , ya sea colaborando con otros jugadores u obteniendo buena nota en los objetivos fijados.
- La segunda parte, los participantes llevarán a cabo tareas de diferentes estilos de liderazgo en diferentes simulaciones.

Las competencias que se desarrollan en esta actividad son el liderazgo, la comunicación, el trabajo en equipo.

En esta actividad, *vLeader* sigue la siguiente metodología:

- Los alumnos tendrán actividades en solitario y otras en grupos.
- Se han aplicado la estrategia de gamificación por el modo de adquirir los objetivos.
- El método seguido es el de simulación, dado que se simulan acciones lo más reales posibles.

Tendrá el código AC2.

## 3. Co-Existence In Melilla[SSF17]

Esta actividad busca el objetivo de una convivencia entre diferentes culturas. Por ello, se ha buscado una ciudad que convivan diferentes culturas, religiones e idiomas. Esta ciudad es Melilla que cumple con todas estas características.

Las competencias que se recogen son la comunicación, la capacidad de apreciación de la diversidad y multiculturalidad, desarrollo y gestión de proyectos y el emprendimiento social.

Todo esto forma parte de actividades extraescolares del máster de Diversidad cultural. Estas suelen ser las actividades que suelen hacer:

## 2.1. Competencias y actividades

- La primera propuesta es la interacción con los habitantes de Melilla.
- La segunda actividad es hacer un estudio autobiográfico, recogiendo sus vidas, su forma de ver el mundo, cultura, sobre la población en la ciudad.

En esta actividad podemos observar la siguiente metodología:

- Los alumnos tendrán que hacer grupos para poder realizar las actividades. De estas tendrán que hacer entrevistas y seminarios.
- Se ha aplicado la estrategia de aprendizaje basado en proyectos.
- El método seguido ha sido el empírico, como la recogida de información para como parte de las actividades del proyecto.

Tendrá el código AC3.

4. **The Class Representative System Of Trinity College Dublin students[JWO17]**. Este iniciativa forma parte de un proyecto formado por alumnos del Colegio Trinity de Dublin. Corresponde con un sindicato estudiantil. Estos alumnos fuera de la hora lectiva se reunen para hacer actividades como campañas, foros de debate y eventos de todo tipo para poder ayudar a todos los estudiantes.

Una de sus partes es comprometerse y contribuir, participando en las elecciones, donde tienen que demostrar su capacidad de liderazgo y comunicación.

En estas actividades podemos observar la siguiente metodología:

- Los alumnos tendrán que formar grupos para participar en los foros de debate y en las campañas.
- Sigue la estrategia de aprendizaje por debate.
- Sigue un método didáctico.

Las competencias que se recogen son la comunicación, el trabajo en equipo y el liderazgo.

Tendrá el código AC4.

5. **An Experience At The Colegio Mayor Padre Poveda In Madrid[RGM17]**. El proyecto de este colegio universitario se llama "We Count", donde se buscan que los alumnos participan en la resolución de problemas. El punto fuerte de este proyecto, es que a los alumnos se le enseñan diferentes formas de enfrentarse a un problema complejo.

En esta actividad podemos observar la siguiente metodología:

- Se forman grupos pequeños para poder resolver un problema planteado.
- Sigue la estrategia de design thinking .
- Sigue el método de reflexión y acción.

Las competencias que se recogen son la comunicación, la capacidad de resolver problemas o conflictos.

Tendrá el código AC5.

6. **The College Council[RM17]**. El proyecto se desarrolla en la universidad de Valladolid donde se ofrece diferentes actividades para el alumnado creado por el consejo de estudiantes. Este se le conoce con el nombre de College Council, donde a los alumnos se les motiva para participar en:

## *2. Fundamentos y bases tecnológicas*

- Diferentes programas a lo largo de año.
- Difundir su programa para que intervengan la mayor cantidad de alumnos.
- Llevar a cabo proyectos para desarrollar competencias genéricas.

Entre las actividades que desarrollan, las principales fueron la creación de proyectos para la mejora de las competencias, llevar a cabo campañas para que los alumnos se presenten a los organismos de la facultad y poder contribuir en ella, y con ello desarrollando la capacidad de liderazgo.

Entre la metodología seguida podemos destacar los siguiente:

- La mayoría de las experiencias y actividades se hacen en grupo.
- Se sigue la línea de estrategia por aprendizaje por proyectos y por debate.
- El método en función de la actividad.

Las competencias que se desarrollan en esta actividad son el liderazgo, la comunicación, el trabajo en equipo.

Tendrá el código AC6.

7. **Galilean School Of Higher Education**[[Ben17](#)]. Este proyecto iniciado por la universidad de Padova, donde se crea un pequeño de 30 personas llamados "Galilean Student" seleccionados entre los mejores de toda la facultad.

Durante su periodo en la facultad tendrán que hacer cursos y abundantes actividades, en función de la carrera universitaria que estén cursando. Todo esto incluye seminarios, trabajos en grupo, exposiciones, presentarse al consejo estudiantil, etc.

Entre la metodología que sigue la facultad, podemos destacar los siguiente:

- Las actividades pueden ser en grupo o individual.
- La estrategia seguida depende de los cursos o experiencias.
- El método seguido también depende al igual de la estrategia.

Las competencias que se recogen son el trabajo en equipo, el liderazgo, la capacidad de apreciación de la diversidad y multiculturalidad, el desarrollo y gestión de proyectos y el emprendimiento social.

Tendrá el código AC7.

8. **Uppsala Student Nations**[[ALJS17](#)]. La nación de estudiante de Uppsala corresponde con un asociación estudiantil dividido en trece naciones o fraternidades donde la los estudiantes pueden tener acceso a las diferente actividades en el campus de la universidad.

Las competencias que se recogen son la comunicación, el trabajo en equipo, el liderazgo, la resolución de conflictos y el desarrollo y gestión de proyectos.

Cada fraternidad elige entre los suyos a unos representantes para las reuniones donde pueden debatir y defender los intereses de los alumnos.

Entre sus actividades destacan el Varannan Vatten y First of May Dinner donde se busca la cooperación de todas las fraternidades para llevar a cabo decenas de actividades para el alumnado.

La metodología que sigue es la siguiente:

## 2.1. Competencias y actividades

- Es exclusivamente en grupo
- Dado el tamaño gigantesco del proyecto, la estrategia depende de las actividades que hagan .
- De la misma forma que pasa con la estrategia no se pueden aplicar un solo método.

Tendrá el código AC8.

### 9. The Use Of Portfolios And Reflective Logs To Connect Learning Acquired In Formal And Informal Contexts[ASG17].

Este proyecto de la universidad de Padova tiene como objetivo en apoyar a los alumnos para que sean capaces de identificar las competencias que han desarrollado en la facultad(en un contexto formal) y saber aprovecharlas y relacionarlas con el aprendizaje informal y no formal.

Para el proyecto suelen escoger aproximadamente 20 alumnos y tiene una duración de hasta 2 años.

Estas son las actividades que suelen hacer:

- Deben hacer un diario de seguimiento de todas las actividades que hacen en la facultad y identificar las competencias que han adquirido.
- Construir su propio portafolios con las experiencias aprendidas con el apoyo de un tutor.

Las competencias que se recogen son la comunicación, el trabajo en equipo, el liderazgo, la capacidad de apreciación de la diversidad y multiculturalidad, la resolución de conflictos, la resolución y gestión de proyectos y el emprendimiento social.

Tendrá el código AC9.

En el caso del contexto de aprendizaje depende de la flexibilidad que ofrece cada actividad pero todas están apoyadas por instituciones.

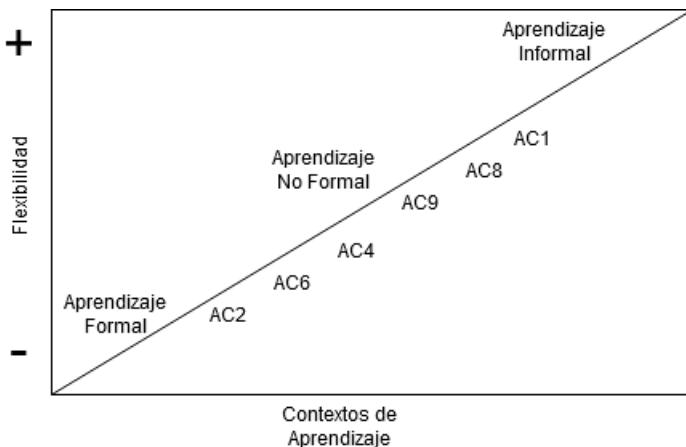


Figura 2.7.: Flexibilidad de las actividades propuestas según Smitter.

## *2. Fundamentos y bases tecnológicas*

### **2.2. Economía Colaborativa**

El concepto de economía colaborativa se ha ido desarrollando a lo largo de estos últimos años y como se presentó como algo tan novedoso, que los expertos no han llegado a ningún consenso.

#### **2.2.1. Conceptos básicos**

El objetivo de este apartado nos ayudará a entender algunos conceptos que se desarrollarán a lo largo de este capítulo.

**Economía colaborativa [Owy15]** Modelo económico en el que las tecnologías comúnmente disponibles permiten a las personas obtener lo que necesitan unas de otras.

**Consumo colaborativo [Che20]** Es el uso compartido de un bien o servicio por un grupo a cambio de una compensación no necesariamente monetaria pactada entre las partes.

**Producto [pro]** Bien tangible u objeto que ofrece una empresa a los consumidores.

**Servicio [pro]** Bien intangible o conjunto de actividades que se ofrece, generalmente, como complemento de un producto.

**Comprador[com]** Se define como la persona que adquiere un producto o servicio por una cantidad de dinero. Es pieza clave en el intercambio comercial.

**Consumidor[com]** Es aquella persona que hace uso del producto comprado ya sea por sí mismo o por un tercero, sin necesariamente estar en contacto directo con el punto de venta. También se puede pensar que es quien vive la experiencia de consumo de aquello que se adquiere.

**Cliente[rae]** La persona que compra habitualmente en la misma empresa (consumidor habitual).

**Proveedor[Gal]** Aquella persona física o jurídica que provee o suministra profesionalmente de un determinado bien o servicio a otros individuos o sociedades, como forma de actividad económica y a cambio de una contra prestación. Un proveedor también ofrece servicios profesionales intangibles como horas de consultoría, formación, etc.

**Prosumidor[noz19a]** Es un consumidor que se involucra en la producción de los servicios y los bienes que adquiere, consume o utiliza.

**Usuario[rae]** La persona que disfruta habitualmente de un servicio o del empleo de un producto.

**Relación peer to peer(P2P) [eco20]** Es un modelo de negocio, basado en el intercambio o venta, ya sean productos y/o servicios entre particulares y/o empresas pero al mismo nivel.

**Relaciones contractuales(B2C) [eco20]** Hacen referencia al comercio de empresas a particulares.

**Relaciones B2B [eco20]** Son transacciones comerciales entre una empresa y otra empresa.

**Relaciones C2C [eco20]** Consiste en facilitar el encuentro y vincular a personas con necesidades o intereses compatibles, es decir, no ofrece un producto o servicio propio, sino que la empresa se convierte en un intermediario entre personas.

**Plataformas colaborativas [pla19]** Hacen referencia a aquellos espacios exclusivamente digitales que tienen el objetivo de unir, en un mismo lugar, a profesionales para fomentar su colaboración e interacción.

**Plataformas digitales [Gir19]** Espacios en Internet que permiten la ejecución de diversas aplicaciones o programas en un mismo lugar para satisfacer distintas necesidades.

**Crowdfunding [viv20]** Consiste en una red de financiación colectiva, normalmente online, que a través de donaciones económicas o de otro tipo, consiguen financiar un determinado proyecto a cambio de recompensas, participaciones de forma altruista.

**TICs [tic20]** Son tecnologías que utilizan la informática, la microelectrónica y las telecomunicaciones para crear nuevas formas de comunicación a través de herramientas de carácter tecnológico y comunicacional, esto con el fin de facilitar la emisión, acceso y tratamiento de la información.

**Software libre [Mej19]** Software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

**Información [yP99]** Es un conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), y que por lo tanto son de utilidad para quién debe tomar decisiones, al disminuir su incertidumbre.

## 2.2.2. Origen y concepto

Para poder hablar de economía colaborativa debemos entender antes que es el consumo colaborativo. Consisten en darle definición a la práctica de acciones que llevamos haciendo con nuestros amigos o familiares como compartir el coche para ir al trabajo, prestarse dinero en vez de que pidan un préstamo en el banco o regalar la ropa. Por eso podemos definirlo como el cambiar el poseer bienes por compartirlos, cambiar la necesidad de maximizar el consumo por la de maximizar la satisfacción [?].

Por ello podemos definir la economía colaborativa como un modelo económico en el que las tecnologías comúnmente disponibles permiten a las personas obtener lo que necesitan unas de otras. Ahora que entendemos el concepto, deberíamos hacernos la siguiente pregunta. ¿por qué se está popularizando tanto este modelo económico y desplazando los modelos clásicos?

Las economías actuales están basadas en que la sociedad, oferta o consume bienes y servicios para satisfacer sus necesidades cotidianas. Estas tienen su límite en el poder adquisitivo del que disponga la población. Dado que la necesidad de consumo, que se ha ido incrementando de forma acelerada en las sociedades occidentales, y el poder adquisitivo no ha crecido de una forma proporcional sino que el poder adquisitivo se ha ido ralentizando.

Por ello, la economía colaborativa intenta subsanar estos problemas, buscando remodelar la situación actual del modelo de mercado, reordenando el modo en el que el consumidor satisface sus necesidades.

## *2. Fundamentos y bases tecnológicas*

No se considera un nuevo modelo de mercado, dado que antes de la aparición de la moneda, la forma de comerciar era el trueque y este se asemeja a la filosofía de la economía colaborativa [noz19b].

### **2.2.3. Pilares**

Anteriormente, hemos hecho un acercamiento sobre qué es la economía colaborativa y como la situación económica y el desarrollo de las tecnologías de la información hizo que surgiera, pero no es una respuesta sencilla ni inmediata. Todavía no se considera un sector definido y claro como es el caso de los modelos de economía tradicionales. Los primeros acercamientos para establecer una teoría y modelo se hicieron en mano de la escritora Rachel Botsman [Bot20] y más tarde por el analista industrial Jeremiah Owyang [Owy16].

A pesar de todo esto, actualmente podemos dar unas ideas fundamentales que sustentan este modelo económico[noz19c]:

1. **Esta práctica no conduce a la adquisición o propiedad de bienes y servicios, sino a un uso compartido.** Como hemos mencionado anteriormente, la idea de propiedad es sustituida por el acceso a esos bienes o servicios. Esto nos permite ,que acceder a estos nos sea por un precio menor, además de evitar importantes costes asociados a la propiedad de ciertos bienes.
2. **Las plataformas hacen de intermediario por medios electrónicos una gran cantidad de prestadores de servicios y los usuarios.** Aunque siempre han existido prácticas cercanas a la filosofía de la economía colaborativa, la configuración actual de este modelo no se podría explicar sin el uso de las plataformas digitales. Gracias a esto, se ponen en contacto constante entre proveedores de servicios y usuarios.
3. **El objetivo común es optimizar los bienes y servicios mediante su uso compartido.** Este modelo económico intenta desplazar la sociedad de consumo instalada actualmente en la sociedad, dado que es consciente de que los recursos son limitados y no todos pueden acceder a ellos. Este suele ser una influencia de la economía circular.
4. **La relación triangular establecida sean primordialmente pares(P2P) y nunca relaciones contractuales(B2C).** Otro de los puntos fuertes de este modelo es que las relaciones que se establecen son relaciones entre iguales o P2P. Este apartado se desarrollará más adelante .

### **2.2.4. Tipos**

El modelo de economía colaborativa se puede aplicar se puede aplicar a todos los ámbitos de la vida cotidiana. Tan solo es necesario abstraer el problema que tenemos y comprobar si es aplicable el modelo. Estos ámbitos se han podido clasificar en cuatro tipos.

- **Conocimiento abierto:** El conocimiento abierto son todas aquellas modalidades que promueven la difusión del conocimiento sin barreras legales o administrativas. Representa los cimientos de la economía colaborativa y afecta al resto de sus formas[Est16]. Todos los programas de código libre, software libre, las webs sobre apuntes o las aplicaciones gratuitas que se pueden utilizar sin problemas legales son un ejemplo de cómo entre todos se puede a generar una gran cantidad de conocimientos al alcance de cualquier persona.

- **Consumo Colaborativo:** El consumo colaborativo ya lo habíamos comentado como unos de los primeros en surgir en la práctica. Desarrollando los fundamentos de la economía colaborativa hasta nuestros días, este ha sido encuadrado en un tipo dentro del modelo.

Por lo tanto ahora podemos decir que el consumo colaborativo utiliza plataformas digitales a través de las cuales los usuarios se ponen en contacto para intercambiar bienes o artículos, casi todos de forma gratuita y altruista [Wah17].

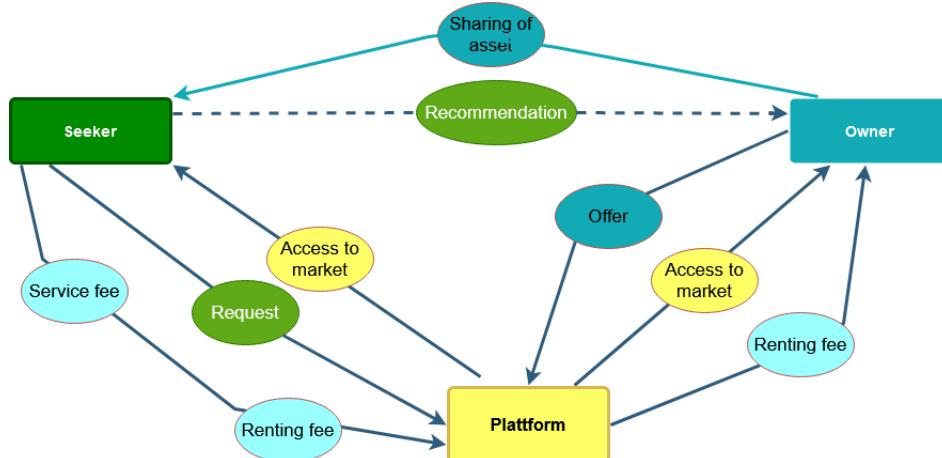


Figura 2.8.: Esquema del consumo colaborativo. [Wah17]

- **Producción Colaborativa:** Se trata de un movimiento llamado “Maker o Do it yourself” que trata de promover la difusión de proyectos de todo tipo. Estos proyectos pueden llevarse de forma sostenible, llegando a crear dispositivos sofisticados como puede ser dispositivos electrónicos. La mayoría de los productos creados bajo este movimiento son de código abierto, ya que cualquiera puede acceder y crearlos utilizando la documentación y los manuales disponibles[MAK12].
- **Las Finanzas Colaborativas:** Corresponde con microcréditos, préstamos, ahorros, donaciones y vías de financiación. Los usuarios se ponen en contacto para satisfacer necesidades en cualquiera de estos aspectos. El mejor ejemplo lo vemos en el crowdfunding, modelo de financiación para aquellos que deseen aportar capital a ciertas iniciativas de forma desinteresada[Bal11].

### 2.2.5. Modelo peer to peer

A lo largo de estos capítulos desarrollados, hemos visto diferentes factores sociales, económicos y tecnológicos que han contribuido al nacimiento de la economía colaborativa. Pero uno de los más importantes es el cambio de relación entre individuos, usuarios, consumidores, profesionales y empresas.

La economía tradicional está cimentada principalmente en dos tipos de relaciones. Por un lado, las relaciones B2B (business to business) y por otro las relaciones B2C (Business to Customer).

## *2. Fundamentos y bases tecnológicas*

Las primeras están asociadas a las relaciones entre profesionales, como puede ser el caso de un distribuidor y un comercio minorista en el ámbito de bienes, o empresas que proveen de contenidos web a otras, ya sea a través de una web para esa empresa o entradas en un blog, tweets, trabajo de posicionamiento web o de redes sociales, portales promocionales en Internet en el en el ámbito de servicios [B2C20].

Las segundas están asociadas a las relaciones entre los profesionales y los consumidores, ya sea la relación de un comercio minorista y las personas que acuden para proveerse de bienes o el proveedor de un servicio de telefonía y sus correspondientes consumidores.

En estas relaciones, vemos que están entrelazadas y gracias a los intermediarios logramos conectarlas como si se tratara de una cadena. En estas relaciones se fue incrementado el poder de los profesionales frente a los consumidores provocando un desequilibrio. A este último se le cedió ciertos derechos para evitar situaciones de abuso.

La economía colaborativa rompe con todo estos esquemas o postulados clásicos y sobretodo con el fuerte desequilibrio provocado en las últimas décadas. Por ello, surgen dos nuevos tipos de relaciones a lo que llamaremos modelo P2P (peer to peer) y el modelo C2C (Consumer to Consumer)[P2P20].

El modelo peer to peer principalmente es un modelo de negocio que, gracias a las plataformas, los usuarios intercambian archivos, datos e información. La relación aquí sitúa a todos por igual , los dos que establecen la interactúan entre sí, pueden ser al mismo tiempo cliente o servidor , lo que ha provocado una gran revolución en los derechos del consumo. Ejemplo de ello podemos encontrarlo en los software como bitTorrent y Ares.

Por último, el modelo C2C podemos considerarlo en una evolución del primero donde los usuarios pasan a ser consumidores manteniendo la relación de igualdad e introduciendo un tercer elemento que corresponde con la plataforma colaborativa. El principal cometido de este modelo es facilitar la comercialización de productos entre particulares. Un ejemplo de ello sería airbnb, donde se cobran comisiones por prestar servicios a proveedores de inmuebles [noz19d].

### **2.2.6. Plataformas colaborativas**

Las tecnologías de la información han revolucionado muchos aspectos de nuestras vidas. Uno de los más importantes es la manera de trabajar, comprar, vender y hacer negocios. A ello se han sumado, en años recientes, las plataformas digitales que ponen en contacto a oferentes y demandantes de productos y servicios, conformando a lo que hemos llamamos plataformas colaborativas.

Aunque la función principal de las plataformas colaborativas no deja de ser la de intermediario en la oferta y demanda en los mercados tradicionales, pero con el desarrollo de las TICs, se ha creado un nuevo escenario o mercado en el que ahora la oferta y la demanda se realiza de forma masiva en estas plataformas[PTC19].

Todo esto provoca que la relación de intermediario o mediación ya no se realiza para un sujeto en concreto, sino para un gran grupo de proveedores y demandantes, provocando que la plataforma no se guíe por la satisfacción de unos pocos, sino de la imparcialidad en relación a todos los sujetos que prestan o reciben servicios a través de ella.

Estas plataformas presentan la siguientes funciones mínimas para considerarse una plataforma colaborativa[noz19e]:

- 1. Crear un mercado en línea.** Como hemos hablando anteriormente, la función de la plataforma es la construcción de un mercado en línea que competirá con otras

plataformas y con el mercado físico.

2. **Facilitar redes de comunicación.** Las plataformas deben tener un sistema de comunicación e interconexión entre usuarios. Estos al mantener una comunicación prácticamente continua entre los individuos que recurren de un modo u otros a la plataforma proporciona una gran flexibilidad, que los modelos de negocios clásicos no pueden aportar.
3. **Ofrecer un servicio de geolocalización.** El desarrollo de las tecnologías móviles, como es el caso de los smartphones, acompañado del potencial desarrollo de los sistemas GPS ha provocado que las plataformas se beneficien de sus sistemas de geolocalización, provocando que la información de los usuarios puedan gestionarse y consultarse en tiempo real.
4. **Gestionar los pagos.** Entre las funciones principales, no podemos olvidar la de ofrecer un conducto de pago seguro para que los usuarios puedan proceder al pago de los servicios solicitados de manera eficaz y segura.



## 3. Análisis y modelado

En este capítulo nos centraremos en el análisis de los diferentes actividades estudiadas en el proyecto Dare+ y Tuning. Con los conocimientos que hemos adquirido en el capítulo 2, podremos buscar los puntos comunes y construir la estructura principal de nuestro proyecto. Por una parte vamos a maximizar el mayor número de actividades para construir un patrón que siguen estas actividades y llevaremos a cabo el análisis de construyendo una configuración de estas.

### 3.1. Análisis de actividades

#### 3.1.1. Selección inicial de actividades

Nuestro objetivo es buscar y construir un módulo con el mayor conjunto de actividades, es decir, maximizar las competencias cubiertas. A este conjunto que tenemos de actividades les extraemos las características comunes buscando patrones.

Para poder hacer todo esto, vamos a organizar las actividades en función del máximo de competencias que cubre. Todo esto, se volverá a hacer con las otras actividades hasta cubrirlas todas posibles, formando cuadrados.

Vamos a partir de una tabla principal donde se resumen todas las actividades y competencias descritas en el capítulo 2. Para comprender que significa cada código, se debe consultar el capítulo 2.1.5 y 2.1.6.

Actividades	CO1	CO2	CO3	CO4	CO5	CO6	CO7
AC1	x	x	x			x	x
AC2	x	x	x				
AC3	x			x		x	x
AC4	x	x	x				
AC5					x		
AC6	x	x	x				
AC7		x	x	x		x	x
AC8	x	x	x		x	x	
AC9	x	x	x	x	x	x	x

Figura 3.1.: Matriz de actividades y competencias

Para esto disponemos de una operación elemental, intercambiar filas y columnas ordenándose hasta encontrar el cuadrado de mayor tamaño. Aplicando el método de eliminación de *Gauss-Jordan* con algunas restricciones.

Tras aplicar una serie de transformaciones llegamos a un cuadrado de 3x3.

Para nuestro modelo, lo ideal sería encontrar un subconjunto homogéneo con el mismo número de filas y columnas, que se distinga de otros subconjuntos, para poder trabajar con él. En este caso, no es posible encontrar un subconjunto 4x4 en nuestra matriz por la siguiente razón [Mir21].

### 3. Análisis y modelado

Actividades	CO3	CO2	CO1	CO6	CO5	CO4	CO7
AC9	x	x	x	x	x	x	x
AC8	x	x	x	x	x		
AC1	x	x	x	x			x
AC6	x	x	x				
AC4	x	x	x				
AC2	x	x	x				
AC7	x	x		x		x	x
AC5					x		
AC3			x	x		x	x

Figura 3.2.: Matriz de actividades y competencias reordenada

Tenemos que buscar filas que tengan al menos 4 unos y columnas que tengan al menos 4 unos (y reordenarlas). Eso descarta las filas 2, 4, 5 y 6 y las columnas 4 y 5. Con esto tendríamos:

Actividades	CO3	CO2	CO1	CO6	CO7
AC9	x	x	x	x	x
AC8	x	x	x	x	
AC1	x	x	x	x	x
AC7	x	x		x	x
AC3			x	x	x

Figura 3.3.: Matriz con al menos cuatro unos

Como lo que buscamos es una matriz de 4x4, la última fila no puede estar.

Actividades	CO3	CO2	CO1	CO6	CO7
AC9	x	x	x	x	x
AC8	x	x	x	x	
AC1	x	x	x	x	x
AC7	x	x		x	x

Figura 3.4.: Matriz resultante

Pero aquí hay sólo tres columnas con cuatro unos. Por tanto, no se puede formar un cuadrado 4x4.

Dado que por esta razón, no podemos construir nuestro modelo con el caso ideal, vamos a hacerlo a partir de dos subconjuntos de nuestra matriz.

Como veremos, nuestros dos subconjuntos tienen la particularidad de que son la intersección del subconjunto 3x3 y muy cercano al 4x4 del homogéneo del que se habló.

Nuestro primer subconjunto es el siguiente:

Este subconjunto de 4x3 de la figura 3.5 que intervienen las competencias CO3(Liderazgo), CO2(Trabajo en equipo), CO1(Comunicación) y CO6(Desarrollo y gestión de proyectos), y las actividades AC9(The use of portfolios and reflective logs to connect learning acquired in formal and informal contexts), AC8(Uppsala student nations) y AC1(Entrepreneurs to change the world).

En estas tres actividades podemos encontrar en común:

Actividades	CO3	CO2	CO1	CO6	CO5	CO4	CO7
AC9	x	x	x	x	x	x	x
AC8	x	x	x	x	x		
AC1	x	x	x	x			x
AC6	x	x	x				
AC4	x	x	x				
AC2	x	x	x				
AC7	x	x		x		x	x
AC5					x		
AC3			x	x		x	x

Figura 3.5.: Primer subconjunto del modelo

- Se forman grupos de 4 o 5 personas.
- Aunque todos parte de una base institucional en un contexto formal, su objetivo es relacionarlo con el aprendizaje informal.
- En todos se crean proyectos para poder desarrollar una serie de competencias genéricas.
- Las personas que participan pueden ser de países diferentes.
- En todas las actividades se fomenta el debate.
- Se desarrollan en ambientes estudiantiles.
- La actividad se desarrolla en unos pocos días.
- Todos los integrantes del grupo pertenecen a la misma institución.
- Las actividades son monitorizadas por un árbitro.

Podemos sacar como conclusión que la competencia que marca aquí la diferencia es el desarrollo y gestión de proyectos. Las otras competencias están subordinadas a esta, es decir, que para que podemos desarrollar un proyecto, tenemos que tener presente que debe existir en el grupo una buena comunicación y alguien que actúe como líder del grupo. Si nos fijamos en los fundamentos que desarrollamos, corresponde con la definición y algunas características dadas.

Nuestro segundo subconjunto es el siguiente:

Este subconjunto de 3x6 de la figura 3.6, en el que intervienen las competencias CO3(Liderazgo), CO2(Trabajo en equipo) y CO1(Comunicación), y las actividades AC9(The use of portfolios and reflective logs to connect learning acquired in formal and informal contexts), AC8(Uppsala student nations), AC1(Entrepreneurs to change the world), AC6(The college council), AC4(The class representative system of trinity college Dublin students), AC2(Learning through an educational simulator).

En estas seis actividades podemos encontrar en común:

- Se forman grupos de 4 o 5 personas en promedio.
- Parte de una base institucional en contexto formal, pero su objetivo se encuentra en un contexto de aprendizaje informal o no-formal.

### 3. Análisis y modelado

Actividades	CO3	CO2	CO1	CO6	CO5	CO4	CO7
AC9	x	x	x	x	x	x	x
AC8	x	x	x	x	x		
AC1	x	x	x	x			x
AC6	x	x	x				
AC4	x	x	x				
AC2	x	x	x				
AC7	x	x		x		x	x
AC5					x		
AC3			x	x		x	x

Figura 3.6.: Segundo subconjunto del modelo

- El objetivo final es cubrir una serie de competencias genéricas.
- Se desarrollan en ambientes estudiantiles.
- Las actividades se monitorizan por árbitros.
- La actividades se desarrollan en unos pocos días.
- Todos los integrantes pertenecen a una misma institución.

Podemos observar que viendo las diferentes actividades se mantienen algunas características del modelo anterior, pero en este caso, ya no es importante ni gira en torno a la creación de un proyecto, sino que se enfoca en actividades para mejorar una serie de competencias en específico. También podemos decir que las actividades al empezarlas se le lleva a cabo un análisis para poder afrontarlas.

Tras analizar nuestros dos subconjuntos, sería interesante ver el resto del conjunto y poder sacar más conclusiones que pueden complementar nuestro modelo.

Este está formado por las competencias CO5(Resolución de problemas y conflictos), CO4(Capacidad de apreciación de la diversidad y multiculturalidad), CO7(Emprendimiento social) y las actividades AC3(Co-Existence in Melilla), AC5(An experience at the colegio mayor padre Poveda in Madrid), AC7(Galilean school of higher education).

De las tres actividades podemos encontrar lo siguiente:

- Grupos de 30 personas.
- Se realizan actividades culturales como ciencia, cine y literatura.
- Se puntúa a los participantes de la actividad.
- Crean su propia actividad desde cero.
- Duración de 3-4 semanas.
- Asignar un arbitro por cada alumno.
- Actividades multiculturales.
- Crean una encuesta de cada actividad que hacen.

### 3.1.2. Análisis de las actividades recogidas

A raíz de las conclusiones sacadas, hemos podido extraer y construir una serie de configuraciones de cada una de las actividades que forman parte de nuestro conjunto principal. Dado que nuestro modelo está formado por dos subconjuntos, escogeremos el segundo para poder mostrar un mayor número de ejemplos de configuraciones.

Todas las actividades y competencias descritas en el capítulo 2. Para comprender que significa cada código, se debe consultar el capítulo [2.1.5](#) y [2.1.6](#).

Partiendo de la información disponible, intentaremos analizar y parametrizar e iremos construyendo diferentes configuraciones con las actividades estudiadas, para satisfacer el objetivo de la actividad:

- 1. AC1. Entrepreneurs To Change The World.** Tal y como comentamos en el capítulo [2](#), esta actividad mezclan una serie de parámetros, por lo que podríamos construir dos o más configuraciones como si se pudieran distinguir diferentes actividades dentro de una actividad. Esto es posible dado que se hacen diferentes tipos de actividades y cada una con diferente

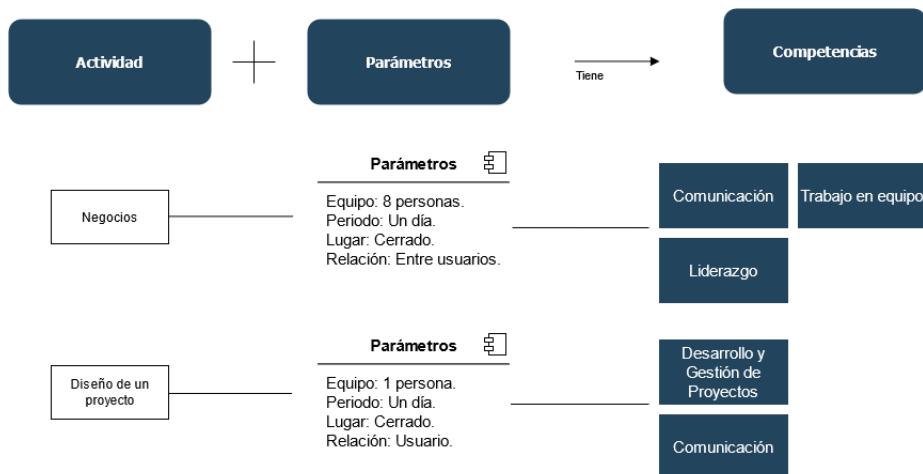


Figura 3.7.: Configuración de dos de las sesiones.

- 2. AC2. Learning Through An Educational Simulator.** Este caso, el uso de la realidad virtual combinándolo con actividades relacionadas con los negocios por lo que presentaría el siguiente esquema.

### 3. Análisis y modelado

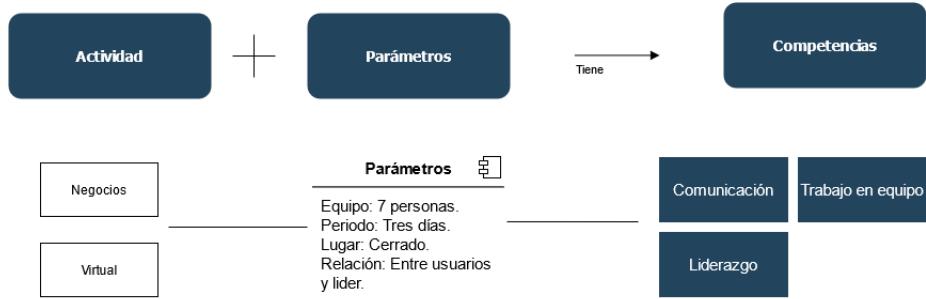


Figura 3.8.: Configuración de la actividad.

### 3. AC4.The Class Representative System Of Trinity College Dublin Students

Este caso es bastante especial, ya que presenta un periodo de desarrollo bastante largo para que solo se enfoque en el debate además de un gran número de personas que se. Esta actividad presenta la siguiente configuración.

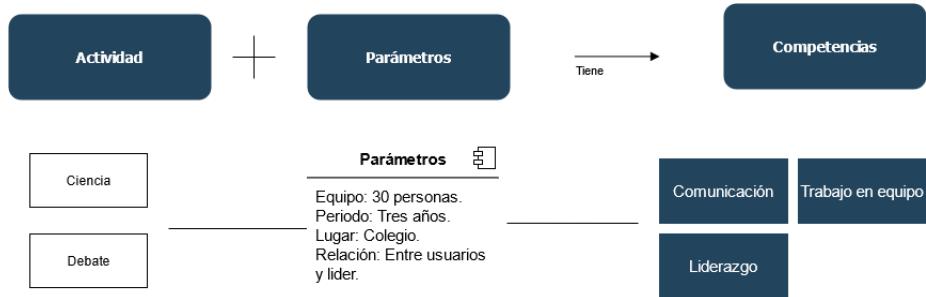


Figura 3.9.: Configuración de la actividad.

### 4. AC6.The College Council

Existen un cierto paralelismo con el anterior en el solo se diferencia en el tipo de actividad y algún parámetro. Esta actividad presenta la siguiente configuración.

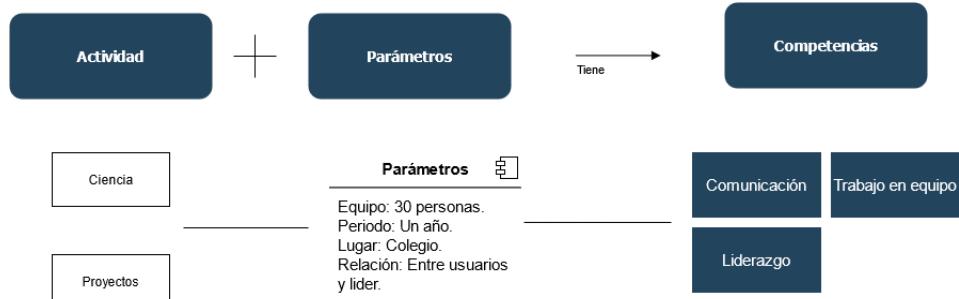


Figura 3.10.: Configuración de la actividad.

### 3.1. Análisis de actividades

5. AC8.Uppsala Student Nations Si leemos el capítulo podríamos darnos cuenta, que hay competencias que faltan, dado que uno de los objetivos es la conexión entre diferentes culturas, además de que no se lleva a cabo ningún proyecto.

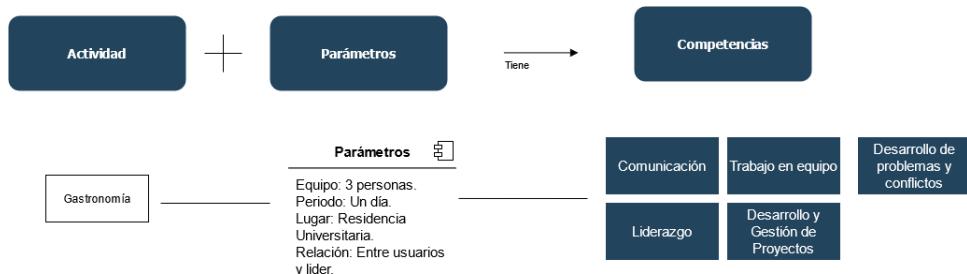


Figura 3.11.: Configuración de la actividad.

### 3. Análisis y modelado

6. AC9.The Use Of Portfolios And Reflective Logs To Connect Learning Acquired In Formal And Informal Contexts En este caso se presenta una actividad central en el que se complementa con otras actividades, es decir, para construir el portafolios se logra a través de otras actividades más básicas o como complemento.

Esta actividad presenta la siguiente configuración.

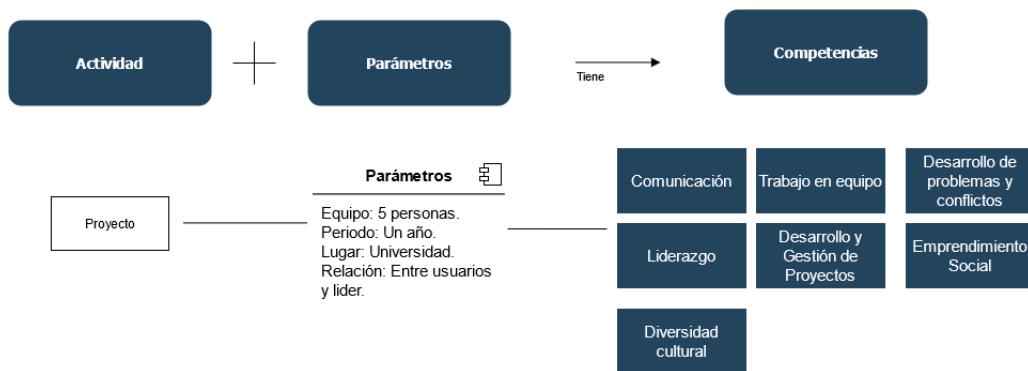


Figura 3.12.: Configuración de la actividad.

Podemos ver que en algunas de las configuraciones presentan algunos estructuras diferentes al resto. Hemos podido ver que hay algunas actividades que presentan dos tipos de actividades diferentes para dar lugar a una específica. Algunas de ellas presentan un periodo largo de hasta más de un año. En otras, vemos que hay actividades dentro de actividades pero no se recoge ninguna estructura, lo que da lugar a una confusión. Otro problema que presentan, es que se rigen por estructuras rígidas para poder completar esas competencias.

También podemos observar es que muchas de estas actividades pueden satisfacerse con otra configuración como es el caso del lugar o del diferente tamaño del grupo.

Además son actividades que se desarrollan a lo largo de un periodo largo y tiene una complejidad mayor para poder parametrizarlas.

## 3.2. Modelado de los parámetros y valores.

Durante el desarrollo de este capítulo se expondrá el diseño planteado para la construcción de nuestro modelo que satisfaga todas las necesidades que hemos visto anteriormente, además de introducir nuevos elementos.

Nuestro modelo llevado a cabo, aunque podría contemplar más competencias, el proyecto se centra específicamente en las competencias genéricas comentadas en el apartado.

Se hará un recorrido por las decisiones tomadas y cómo se traducen a un diseño conceptual y funcional.

El modelo se compondrá de parámetros flexibles y que se puedan elegir entre varios. Tenemos que tener en cuenta que esta parte irreal, es decir, el usuario podrá componer la parte probable de la actividad.

Para todo esto se ha construido un modelo de información como un modelo de entidad y relación para una base de datos.

### 3.2.1. Parámetros y valores.

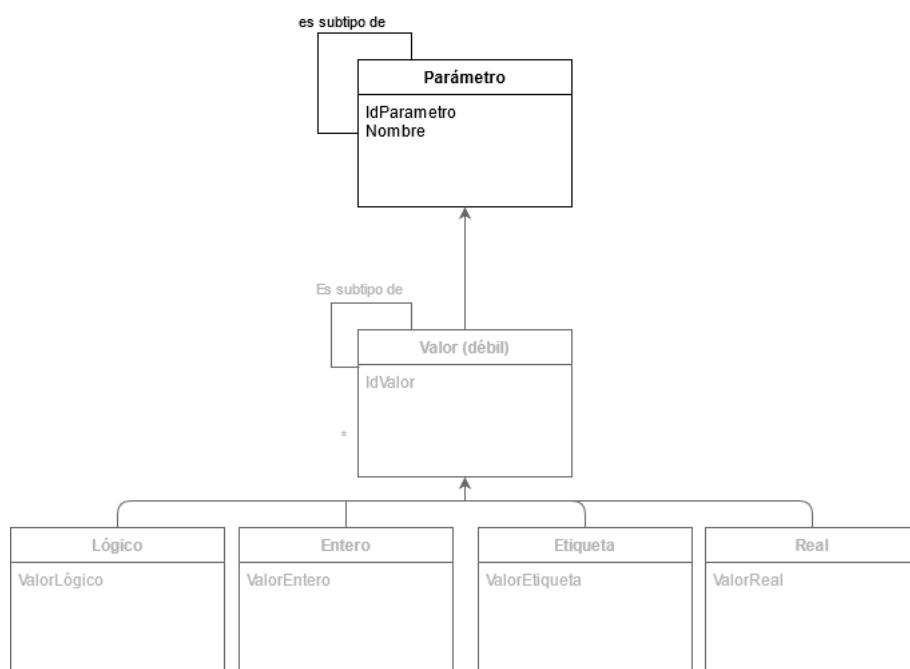


Figura 3.13.: Entidad parámetro.

El primero con la que vamos a tratar es la entidad *parámetro*. Esta estará compuesta por una relación de jerarquía. Dentro de nuestra primera entidad tendremos los parámetros que analizamos en los capítulos anteriores y algunos que hemos añadido a nuestro modelo.

Esta entidad estará descrita por los atributos:

- *IdParámetro*:Corresponde con el identificador del parámetro.
- *Nombre*:Corresponde con el nombre del parámetro.

### 3. Análisis y modelado

Además, esta entidad se relaciona consigo misma estableciendo una relación de jerarquía entre los parámetros, de forma que puede contener subparámetros dentro de un parámetro. Además estos están relacionados entre valores cuantitativos y cualitativos(Esto se explicará en el capítulo siguiente). Una de las ventajas conseguidas es que el diseño nos permite la escalabilidad en el sistema.

En el caso de los Parámetros 3.13, la estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente forma:

idParámetro	Nombre	EsSubtipo
1	Equipo	
2	Tamaño del equipo	1
3	Lugar	
4	Relación	
5	Idioma	
6	Grupo	
7	Tamaño del grupo	6
8	Periodo	
9	Disponibilidad	

Tabla 3.1.: Entidad parámetro

La siguiente entidad dentro de nuestra jerarquía corresponde con *valor* y se especializa mediante una relación de generalización. En ella tendremos todos los parámetros secundarios ordenados además del subtipo del parámetro que corresponda.

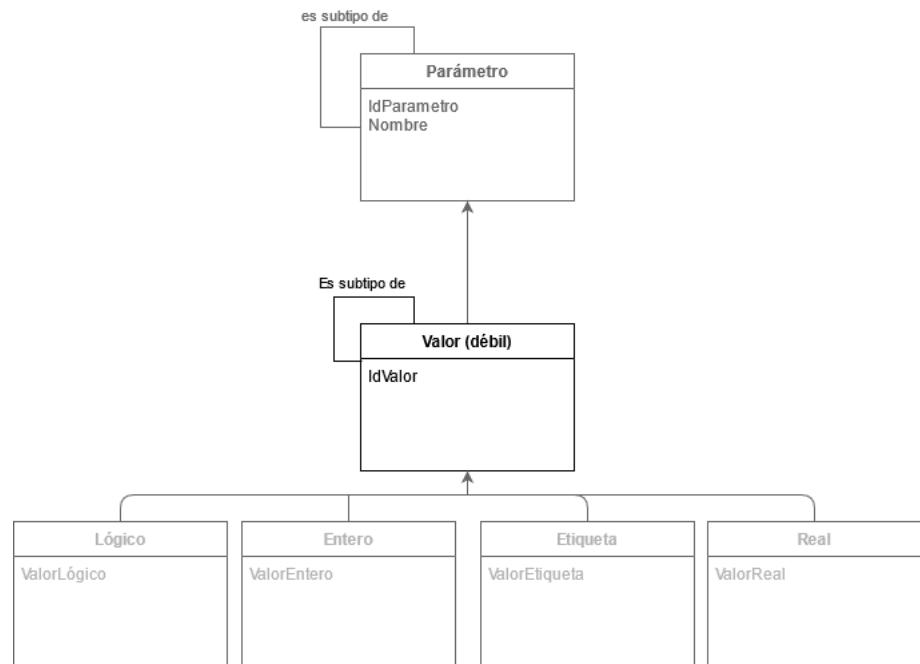


Figura 3.14.: Entidad valor.

### *3.2. Modelado de los parámetros y valores.*

Esta entidad de la figura 3.14 estará descrita por los atributos:

- *idParámetro*:Corresponde con el identificador del parámetro que hereda de la entidad Parámetro.
- *idValor*:Corresponde con el identificador del valor del parámetro. En este caso no corresponde con el valor que tenga sino solo la estructura que contendrá.

<b>idParámetro</b>	<b>IdValor</b>	<b>EsSubtipo</b>
1	1	
1	2	
2	1	
2	2	
3	1	
3	2	
3	3	1
4	1	
5	1	
5	2	
6	1	
6	2	
7	1	
8	2	

Tabla 3.2.: Entidad Valor

### 3. Análisis y modelado

Las cuatro entidades restantes corresponderían a una relación de especialidad en el que se clasificaría los datos en función del tipo de valor que guardamos. Tendremos almacenado el valor de la entidad *valor débil* en función del tipo de dato que almacenemos.

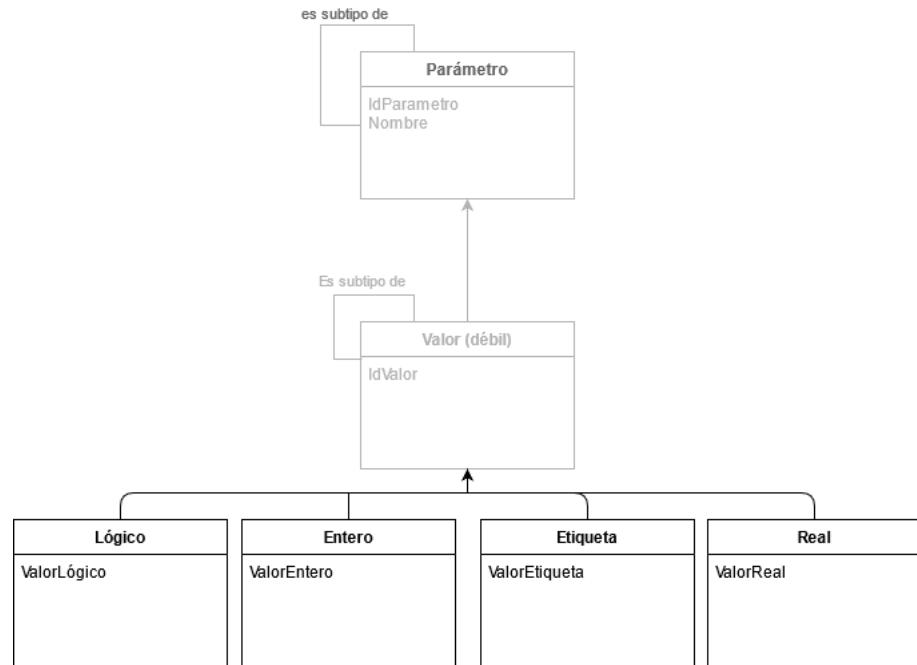


Figura 3.15.: Entidad lógica, entero, etiqueta, real

Estas entidades estarían descritas por los atributos:

- *idParámetro*:Corresponde con el identificador del parámetro que hereda de la entidad Valor.
- *idValor*:Corresponde con el identificador del valor del parámetro. En este caso no corresponde con el valor que tenga sino solo la estructura que contendrá. Esta hereda de la entidad Valor.
- *valorLógico*: Este atributo correspondería con la entidad Lógica.
- *valorEntero*: Este atributo correspondería con la entidad Entero.
- *valorEtiqueta*: Este atributo correspondería con la entidad etiqueta.
- *valorReal*: Este atributo correspondería con la entidad Real.

Todas estas correspondería con el valor que tendría el parámetro.

### 3.2. Modelado de los parámetros y valores.

Un ejemplo de cómo acabaría estructurado la información es esta.

<b>idParámetro</b>	<b>idValor</b>	<b>valorEtiqueta</b>
1	1	Pareja
1	2	Pequeño
3	1	Abierto
3	2	Cerrado
3	3	Casa particular
4	1	Entre usuarios
5	1	Español
5	2	Inglés
6	1	Pequeño
6	2	mediano
8	1	un día

Tabla 3.3.: Entidad Etiqueta 3.15

<b>idParámetro</b>	<b>idValor</b>	<b>valorLógico</b>
9	1	0
9	2	1

Tabla 3.4.: Entidad lógica 3.15

<b>idParámetro</b>	<b>idValor</b>	<b>ValorReal</b>
10	1	3.34
10	2	4.78
10	3	8.89

Tabla 3.5.: Entidad Real 3.15

### 3. Análisis y modelado

idParámetro	idValor	valorEntero
2	1	2
2	2	4
2	3	6
2	4	8
7	1	10
7	2	20
7	3	30
7	4	40

Tabla 3.6.: Entidad Entero 3.15

#### 3.2.2. Tipos de actividades.

En la entidad *tipos de actividades* almacena la información de todos los tipos de actividades recogidos en el capítulo 2. La idea es tener actividades generales y tener subtipos de estas más específicas. El usuario podría combinarlas ya que podría al construir su propia configuración, encontrará lo que realmente busca en su actividad .

La estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente entidad:

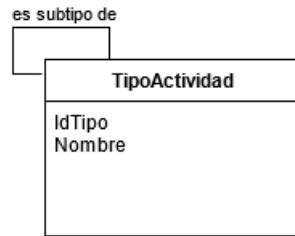


Figura 3.16.: Entidad tipo de actividad.

En esta tabla se almacenará todos los tipos de actividades, lo que supondrá un total de nueve filas. Los campos dentro de la tabla corresponden:

Estos serían nuestros tipos de actividades principales, aunque esta podría ampliarse dado que nuestro sistema es escalable:

1. Deporte
2. Cultura
3. Gastronomía
4. Diversidad cultural
5. Juegos
6. Proyectos
7. Música
8. Ciencia

### 3.2. Modelado de los parámetros y valores.

9. Virtual
10. Naturaleza

Los campos dentro de la tabla corresponden:

- Un identificador que identifica el tipo de actividad.
- El nombre del tipo de actividad.

Además esta entidad se relaciona consigo misma estableciendo una relación de jerarquía entre los sub-tipo(s) de actividades de forma que un tipo de actividad ,puede ser *sub-tipo* de otro tipo de actividad.

#### 3.2.3. Competencias.

Las competencias genéricas seleccionadas para ser probadas y usadas en nuestro sistema son las que se hace mención en el capítulo [2.1.5](#) que corresponden a estas 7 competencias:

Los campos dentro de la tabla corresponden:

1. Comunicación
2. Trabajo en equipo
3. Liderazgo
4. Capacidad de apreciación de la diversidad y multiculturalidad
5. Resolución de problemas y conflictos
6. Desarrollo y gestión de proyectos
7. Emprendimiento social.

El usuario seleccionará las competencias que se van a adquirir en su actividad.

En el caso de las competencias, la estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente tabla:

Competencias
Nombre
IdCompetencia

Figura 3.17.: Entidad competencia.

Los campos dentro de la tabla corresponden:

- El primero corresponde con el identificador de la competencia.
- El segundo corresponde con el nombre de la competencia.

Las competencias Comunicación y Trabajo en equipo se han contemplado como competencias garantizadas por nuestro sistema, dado que sólo se contemplan actividades en grupo y no en solitario.

### 3. Análisis y modelado

#### 3.2.4. Configuración.

La entidad configuración recogerá todas las configuraciones, para nuestras actividades, creadas por el usuario. Será la parte más relevante del modelo de datos, dado que la configuración de la actividad es el objetivo que hemos buscado. La estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente tabla:

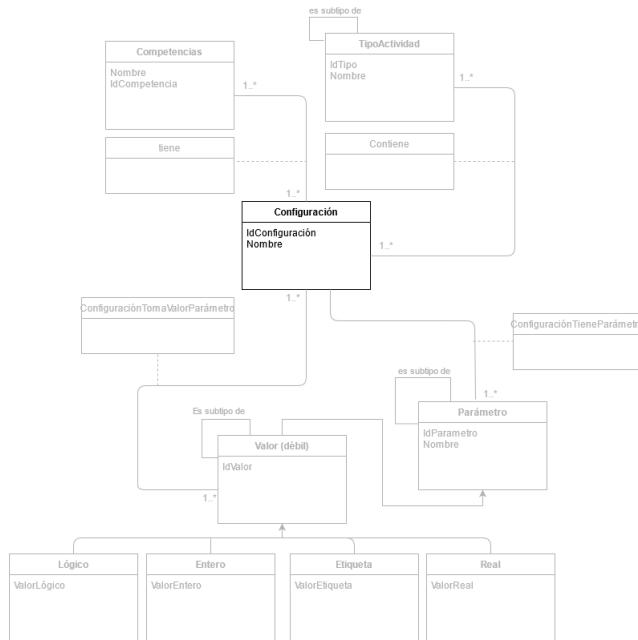


Figura 3.18.: Entidad configuración.

Los campos dentro de la tabla corresponden:

- Un identificador de cada configuración creada.
- El nombre de la configuración.

La importancia que tiene la *configuración* es que va a recoger toda la información para poder crear una actividad y, por ello, debe establecer una relación con las entidades descritas anteriormente.

Por ello vamos establecer cuatro relaciones en torno a la entidad configuración.

### 3.2. Modelado de los parámetros y valores.

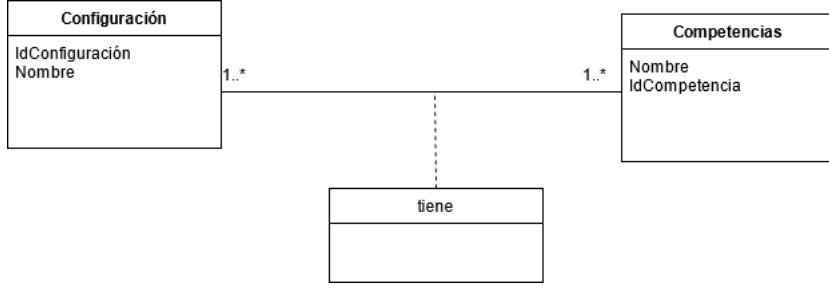


Figura 3.19.: Relación entre configuración y competencia

Estableceremos una relación entre las entidades Configuración-Competencias llamada *Tiene*. En ella el usuario decidirá que competencia satisface su actividad. Como una actividad puede satisfacer varias competencias y viceversa, hemos establecido una cardinalidad de muchos a muchos(n:m). Los atributos por los que sería formado son el identificador de configuración heredado de la entidad configuración y el identificador de la entidad competencia.

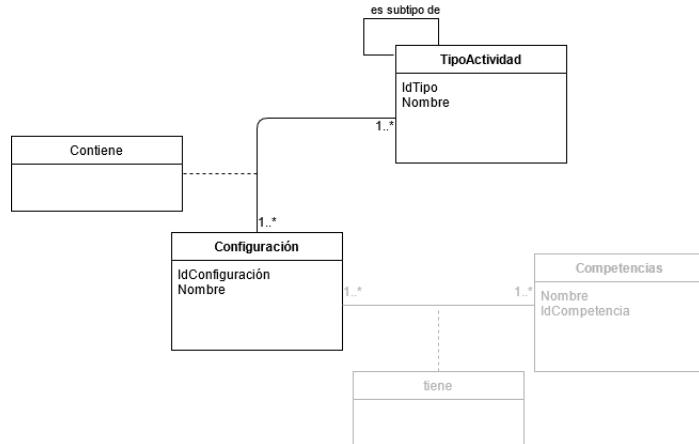


Figura 3.20.: Relación entre configuración y TipoActividad

Estableceremos una relación entre las entidades Configuración-TipoActividad llamada *Contiene*. A través de esta relación, el usuario decidirá de qué tipo es la actividad que está organizando. Como una actividad lo suficientemente compleja ser de varios tipos y puede haber muchas actividades del mismo tipo, hemos establecido una cardinalidad de muchos a muchos(n:m). Los atributos por los que sería formado son el identificador de configuración heredado de la entidad configuración y el identificador de la entidad *TipoActividad*.

### 3. Análisis y modelado

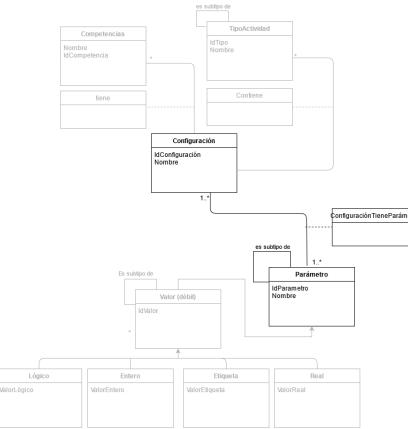


Figura 3.21.: Relación entre configuración y Parámetro

Estableceremos una relación entre las entidades Configuración-Parámetro llamada *ConfiguraciónTieneParámetro*. En ella el usuario seleccionará que parámetros tendrá su actividad. Como una configuración está compuesta por diferentes parámetros y viceversa, hemos establecido una cardinalidad de muchos a muchos(n:m). Los atributos por los que será formado son el identificador de configuración heredado de la entidad configuración y el identificador de la entidad Parámetro.

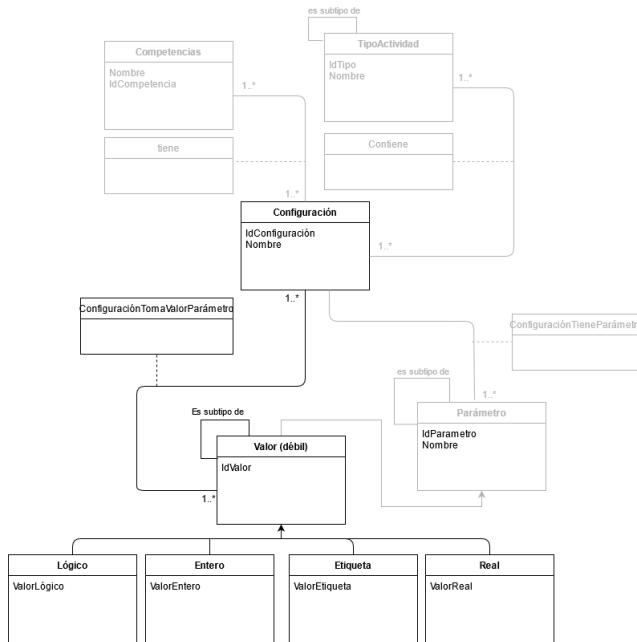


Figura 3.22.: Relación entre configuración y Valor

Estableceremos una relación entre las entidades Configuración-Valor llamada *ConfiguraciónTomaValorParámetro*. El usuario seleccionará el valor de los diferentes parámetros que dispone para su actividad. Como una configuración está compuesta por diferentes valores de

### *3.2. Modelado de los parámetros y valores.*

los parámetros y viceversa, hemos establecido una cardinalidad de muchos a muchos(n:m). Los atributos por los que sería formado son el identificador de configuración heredado de la entidad configuración, el identificador de la entidad Parámetro y un identificador del valor de la entidad Valor.

### *3. Análisis y modelado*

## **3.3. Modelado de las entidades *Usuario, Grupo y Lugar*.**

En nuestra segunda fase vamos a tratar las entidades necesarias para terminar la construcción de nuestro sistema. En esta parte, nos centraremos en los elementos reales, es decir, usuarios, grupos, lugares, y actividad. El objetivo de estas entidades es dar solución a los limitaciones y problemas encontradas en el capítulo anterior.

### **3.3.1. *Usuario*.**

La entidad usuario recogerá a todos los usuarios registrados en nuestro sistema. La estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente tabla:

Usuario
IdUsuario
Nombre
Foto
Biografía
Fecha de nacimiento
Dirección

Figura 3.23.: Entidad Usuario.

Esta entidad estará descrita por los atributos:

- Un identificador que identificará a cada usuario.
- Un nombre elegido por el usuario.
- Una foto del usuario registrado.
- Una biografía para dar una explicación breve del usuario.
- La fecha de nacimiento del usuario.
- Una dirección del estado actual del usuario.

### **3.3.2. *Grupo*.**

La entidad grupo recogerá a todos los grupos creados por los usuarios en nuestro sistema. El grupo es creado por una sola persona y en él se pueden unir diferentes usuarios hasta alcanzar un máximo acordado por el creador del grupo. El objetivo de esta entidad fue pensada, para satisfacer la necesidad de dar una estructura , dado que en los capítulos anteriores veíamos que existían actividades dentro actividades que no presentaba ninguna. A partir de aquí, será el primer paso para que el grupo pueda crear diferentes actividades de forma más coordinada.

La estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente tabla:

### 3.3. Modelado de las entidades Usuario, Grupo y Lugar.

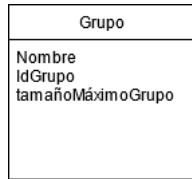


Figura 3.24.: Entidad Grupo.

Esta entidad estará descrita por los atributos:

- Un identificador que identificará a cada grupo.
- Un nombre del grupo.
- Una descripción del objetivo del grupo.
- Un tamaño máximo del grupo.

Entre estas dos entidades descritas se produce dos relaciones:

- La primera relación está relacionada cuando el usuario crea un grupo cualquiera. Necesitamos almacenar quien crea el grupo. Se establece una relación de uno a muchos. El usuario puede crear los grupos que quiera pero el grupo solo es creado solo por un usuario. Estará formado por un identificador del usuario que se ha registrado. Esta correspondería con una clave externa de la entidad Usuario y Un identificador del grupo creado. Esta correspondería con una clave externa de la entidad Grupo. Otro atributo importante sería la fecha de creación.

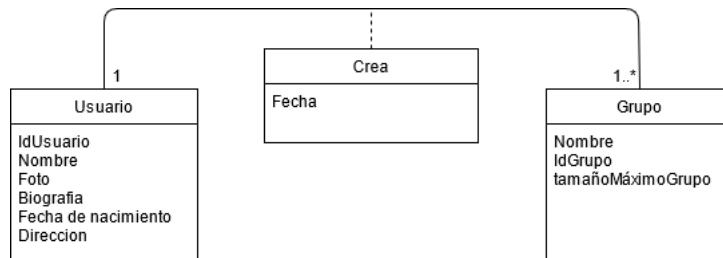


Figura 3.25.: Relación Crea.

- La segunda relación está relacionada cuando el usuario se une a un grupo cualquiera. Necesitamos almacenar a que grupo se une. Se establece una relación de muchos a muchos. El usuario puede unirse a diferentes grupos y en un grupo puede tener diferentes usuarios. Estará formado por un identificador del usuario que se ha registrado. Esta correspondería con una clave externa de la entidad Usuario y un identificador del grupo creado. Esta correspondería con una clave externa de la entidad Grupo. Otro atributo importante sería la fecha de unión al grupo.

### 3. Análisis y modelado

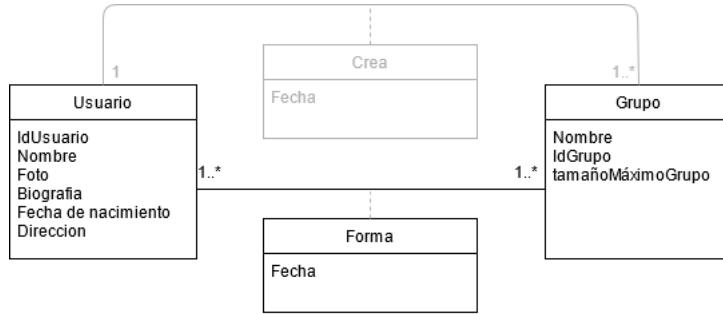


Figura 3.26.: Relación forma.

#### 3.3.3. Lugar.

La entidad lugar recogerá lugares físicos que han sido registrados por usuarios. Estos lugares físicos corresponde con propiedades privadas que ponen su espacio al público y estos podrán ser usados en las actividades. La estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente tabla:

Lugar
Nombre IdLugar Localización

Figura 3.27.: Entidad Lugar.

Esta entidad estará descrita por los atributos:

- Un identificador que identificará de cada lugar que se ha registrado.
- Un nombre del lugar.
- Una localización exacto del lugar.
- El precio establecido por el usuario.

Entre estas dos entidades Usuario-Lugar se produce una relación:

- Esta entidad nace por la necesidad, comentada anteriormente, de que el usuario pueda ceder su propiedad o simplemente establecer un punto de referencia para realizar la actividad. El usuario podrá registrar tanto lugares como necesite pero el lugar solo se puede registrar a través de un usuario. Estará compuesto por un identificador del usuario que se ha registrado. Esta correspondería con una clave externa de la entidad Usuario y un identificador del lugar. Esta correspondería con una clave externa de la entidad Lugar. Otro atributo sería La fecha de registro del lugar.

### 3.3. Modelado de las entidades Usuario, Grupo y Lugar.



Figura 3.28.: Relación cede.

### 3. Análisis y modelado

## 3.4. Modelado de la actividad.

### 3.4.1. Actividad.

La entidad actividad recogerá a todas las actividades creadas por los usuarios en nuestro sistema. La actividad es creada por una sola persona, y que anteriormente ha creado un grupo, es decir, la actividad está subordinada al grupo creado, en el se pueden unir diferentes usuarios del grupo hasta alcanzar un máximo acordado por el creador de la actividad. Como se había comentado el grupo podrá tener diferentes actividades hasta satisfacer la necesidad.

La estructura asignada para almacenar toda la información requerida en nuestro sistema depende de la siguiente tabla:

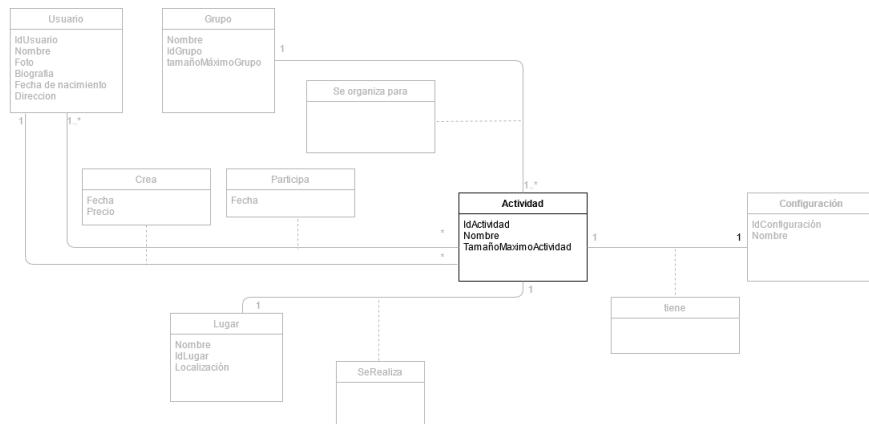


Figura 3.29.: Entidad Actividad.

Esta entidad estará descrita por los atributos:

- Un identificador que representará de la actividad creada.
- Un nombre de la actividad.
- Un tamaño máximo del grupo.
- Una descripción del objetivo de la actividad.

Podemos ver que el eje central de nuestra entidad relación gira en torno a la entidad actividad. Esto es debido a que nuestro sistema se dedica expresamente a gestionar actividades. Por ello, vamos a describir las diferentes relaciones que gira entre ella.

Vemos que hay entre la entidades Usuario-Actividad existen dos relaciones paralelamente a los casos entre el Usuario-Grupo.

- La primera relación está relacionada cuando el usuario crea una actividad cualquiera. Necesitamos almacenar quien crea la actividad. Se establece una relación de uno a muchos. El usuario puede crear muchas actividades que quiera, pero la actividad solo es creada por un usuario. En sus atributos podemos encontrar un identificador del usuario que tiene una clave externa de la entidad Usuario. Un identificador de la

### 3.4. Modelado de la actividad.

actividad que tiene una clave externa de la entidad Actividad. La fecha de creación de la actividad y un precio establecido por el usuario.

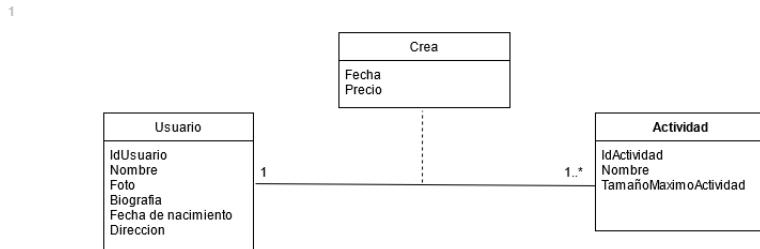


Figura 3.30.: Relación Crea.

- La segunda relación está relacionada cuando el usuario se une a una actividad cualquiera. Necesitamos almacenar a que actividad se une. Se establece una relación de muchos a muchos. El usuario puede unirse a diferentes actividades y en una actividad puede tener diferentes usuarios. En sus atributos podemos encontrar un identificador del usuario que tiene una clave externa de la entidad Usuario. Un identificador de la actividad que tiene una clave externa de la entidad Actividad. La fecha de unión de la actividad.

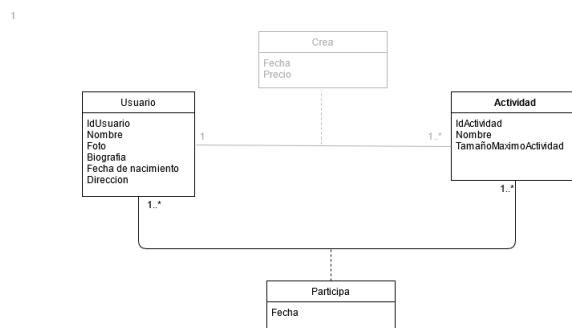


Figura 3.31.: Relación Participa.

Entre la relación de las entidades Grupo-Actividad se debe por la necesidad de esta relación se debe a que habíamos establecido que las actividades estaban subordinadas a un grupo. Con esto solucionábamos la forma de gestionar las actividades dentro de actividades. Un grupo podrá tener diferentes actividades pero una actividad pertenecerá a un solo grupo. Entre los atributos podemos encontrar los identificadores del grupo que es una clave externa de la entidad Grupo y un identificador de la actividad creada, que correspondería con una clave externa de la entidad actividad.

### 3. Análisis y modelado

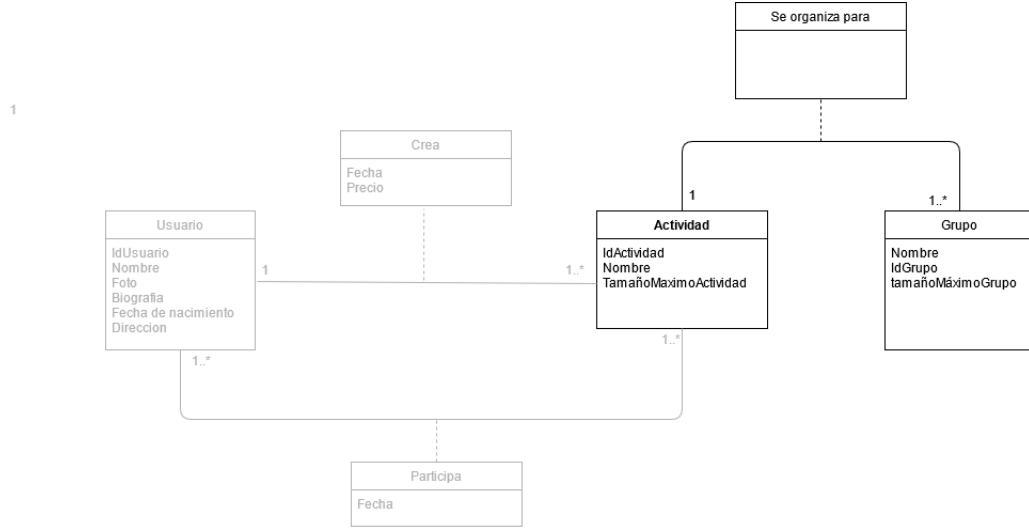


Figura 3.32.: Relación Se organiza para.

Entre la relación de las entidades Lugar-Actividad se establecerá el lugar real que se va a realizar la actividad. Significa que con esta relación nos dice la localización real y no con la configuración que habíamos establecido que correspondería con el tipo de lugar en el que estamos. En este caso una actividad solo correspondería con un lugar y viceversa. Entre los atributos encontramos un identificador del lugar creado, que correspondería con la clave externa de la entidad lugar y un identificador de la actividad que vendría de la clave externa de la entidad actividad.

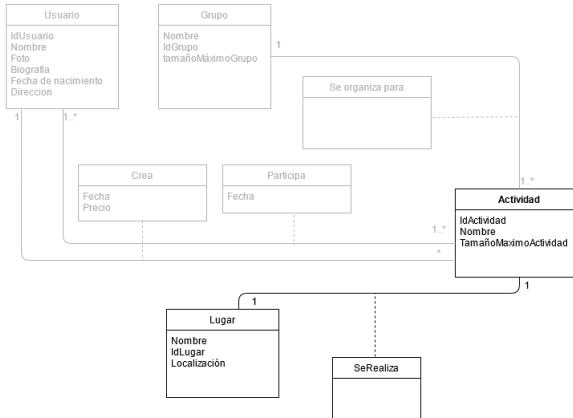


Figura 3.33.: Relación Se realiza.

Entre la relación de las entidades Configuración-Actividad será el lazo de unión entre la actividad y su configuración. Hemos visto en los capítulos anteriores que todas las actividades tenían una serie de características y nosotros los hemos interpretado creando una configuración. Cada actividad tendrá una sola configuración y viceversa. Entre los atributos podemos encontrar identificadores de la configuración, que correspondería con una clave externa de

### 3.4. Modelado de la actividad.

la entidad Configuración y un identificador de la actividad creada, que correspondería con la clave externa de la entidad actividad.

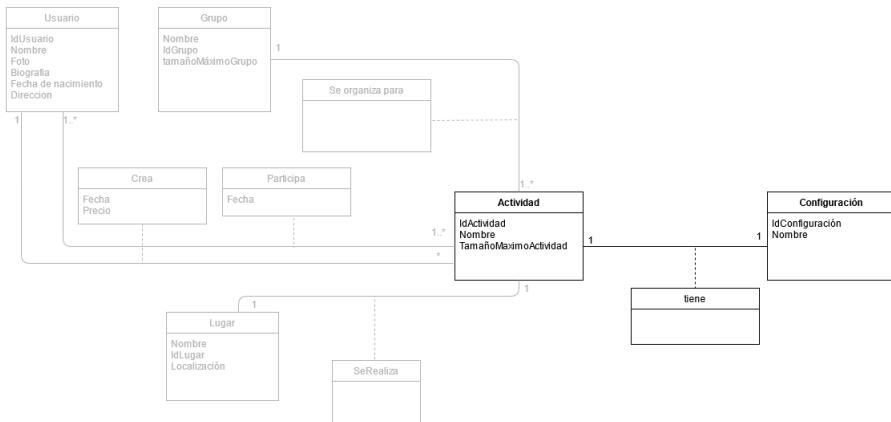


Figura 3.34.: Relación Tiene.

### 3. Análisis y modelado

## 3.5. Modelado de las relaciones entre entidades *Usuario-grupo* y *Parámetro*

Estas relaciones están relacionadas con el uso de nuestra entidad parámetro que dada su propósito podemos usarlo para otras entidades que no sea solo con la configuración. Esto significa que hay entidades como lugar o grupo que establecen relaciones con la entidad parámetro y valor. En estos casos nos es útil utilizar los diferentes parámetros que vamos a establecer.

En la relación entre entidades Grupo-Parámetro se debe a que existen ciertos parámetros que nos son útiles y que tenerlos en esta entidad pueden facilitarnos las cosas. En este caso nos sería útil el idioma que anteriormente ya lo habíamos establecido. Al igual que esta, podríamos hacerlo con cualquiera dado que nuestro sistema es escalable. Encontramos ciertos atributos como un identificador del grupo creado, que sería con una clave externa de la entidad grupo y otro identificador del parámetro, que correspondería con una clave externa de la entidad parámetro.

Por otra parte, paralelamente pasaría lo mismo en la relación Lugar-Parámetro. Encontramos ventajas siguiendo esta línea. Encontramos ciertos atributos como un identificador del lugar creado, que sería con una clave externa de la entidad lugar y otro identificador del parámetro, que correspondería con una clave externa de la entidad parámetro.

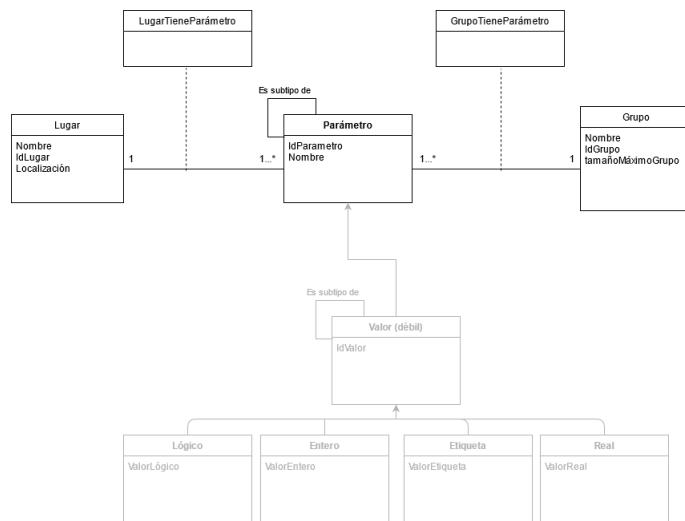


Figura 3.35.: Relación LugarTieneParámetro y GrupoTieneParámetro.

Por ultimo, encontramos las ultimas relaciones pero esta vez con la entidad subordinada Valor. Con estas relaciones estableceríamos los valores que tendría de cada parámetro. En la relación Valor-Grupo encontraríamos los atributos el identificador del grupo, que correspondería con la clave externa del grupo, un identificador de la posición del valor y un identificador del parámetro que hereda de la jerarquía.

En el caso de la relación Valor-Lugar pasaría lo mismo pero en este caso se usaría el identificador del lugar que correspondería con la clave externa de la entidad lugar y en el caso de valor sería igual.

3.5. Modelado de las relaciones entre entidades Usuario-grupo y Parámetro

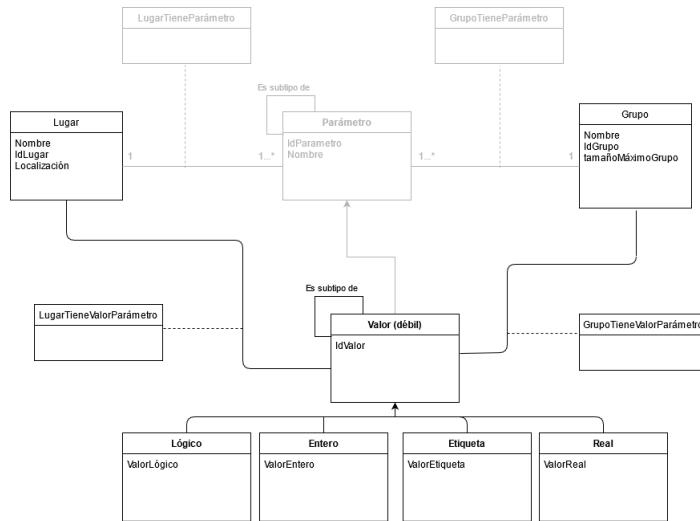


Figura 3.36.: Relación LugarTieneValorParámetro y GrupoTieneValorParámetro.

### 3. Análisis y modelado

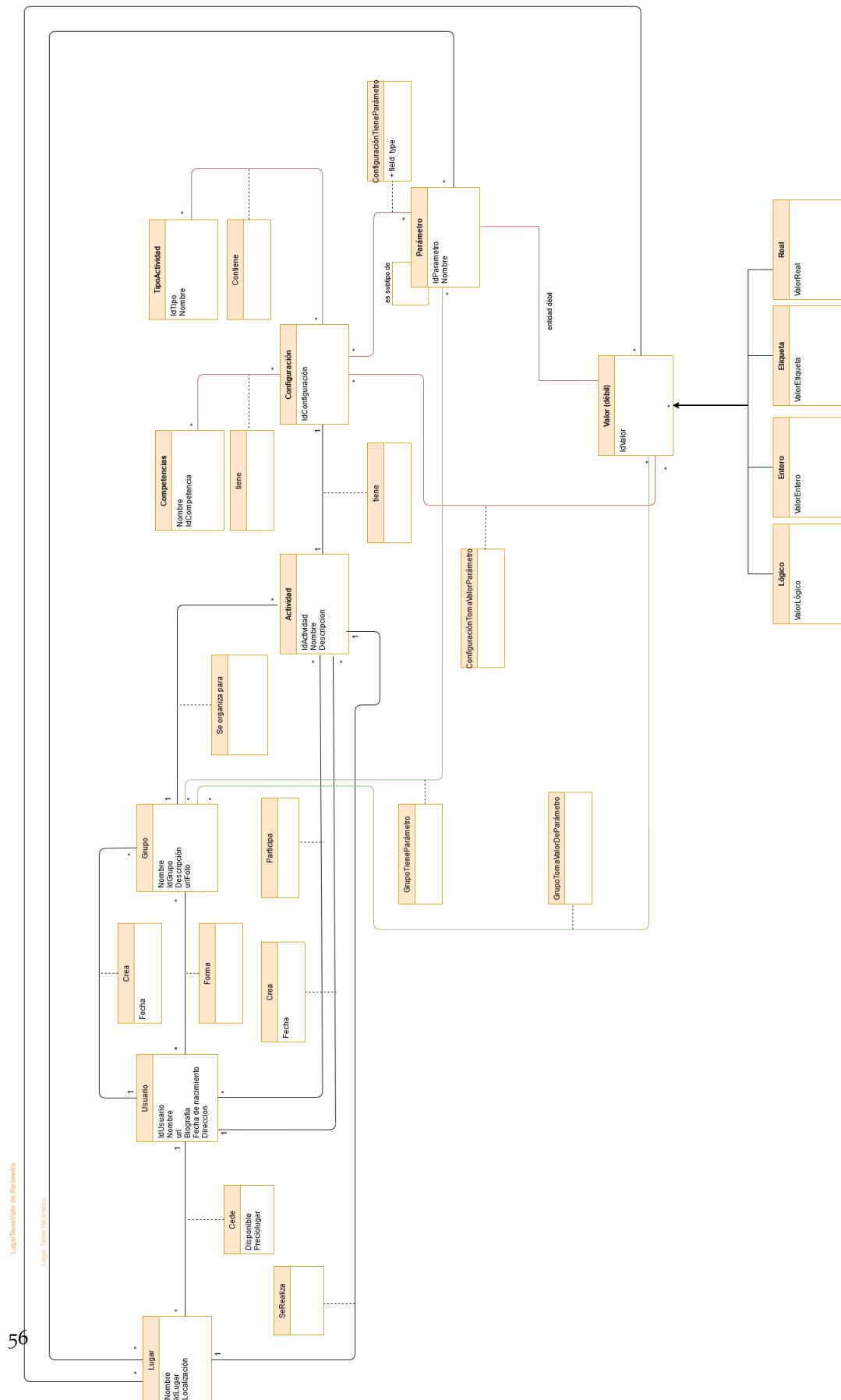


Figura 3.37.: Modelo de información completo.

### 3.6. Ejemplo de funcionamiento del sistema

Para mostrar más claramente lo que se querido mostrar en este capítulo, se ha optado por diseñar casos que muestre como quedaría la información en nuestro sistema. Para facilitar el ejemplo se incluirá solo los atributos más importantes.

Supongamos un ejemplo completo que se usara todo nuestro modelo de información:

Cuatro personas usan nuestro sistema y se registran. Uno de ellos crea un grupo dedicado a la cocina. Otro de ellos solo le interesa en ceder su establecimiento. Los otros dos buscan un grupo dedicado a la cocina. Ademas de crear el grupo quiere crear una actividad para la siguiente semana y estos dos se quieren apuntar.

Tenemos a los siguientes usuarios:

idUsuario	Nombre	Fecha de Nacimiento
0	Juan Molina	2/05/1978
1	Sergio Martín	5/09/1981
2	Ignacio Morillas	9/04/1977
3	Lucía Sánchez	23/11/1968

Tabla 3.7.: Entidad Usuario

Uno de ellos ha cedido su establecimiento para que se pueden hacer actividades.

idLugar	Nombre	Localización	Precio
0	Bar Lucia	Armilla	20 €

Tabla 3.8.: Entidad Lugar

idUsuario	idLugar	Fecha
4	0	17/04/2021

Tabla 3.9.: Relación cede

El usuario pone la disponibilidad del lugar para la actividad.

idLugar	idParámetro
0	9

Tabla 3.10.: Relación LugarTieneParámetro

idLugar	idParámetro	Idvalor
0	9	1

Tabla 3.11.: Relación LugarTieneValorDeParámetro

Uno de los usuarios crea el grupo que quería de cocina.

3. Análisis y modelado

idGrupo	Nombre
o	Grupo de cocina en Granada

Tabla 3.12.: Entidad Grupo

Se le asignan los parámetros del grupo.

idGrupo	idParámetro
o	6
o	5

Tabla 3.13.: Relación GrupoTieneParámetro

idGrupo	idParámetro	idValor
o	6	1
o	5	1

Tabla 3.14.: Relación GrupoTieneValorParámetro

Se crea la relación del usuario y el grupo.

idUsuario	idGrupo	Fecha
o	o	17/4/2021

Tabla 3.15.: Relación Crea

Se unen los usuario el nuevo grupo.

idUsuario	idGrupo	Fecha
1	o	18/4/2021
2	o	19/4/2021

Tabla 3.16.: Relación Forma

El usuario crea una actividad.

idActividad	Nombre
o	Cocina exótica Asiática

Tabla 3.17.: Entidad Actividad

idUsuario	idActividad	Fecha	Precio
o	o	25/4/2021	32€

Tabla 3.18.: Relación Crea

El resto de usuarios se unen a la actividad.

*3.6. Ejemplo de funcionamiento del sistema*

idUsuario	idActividad	Fecha
1	0	25/4/2021
2	0	25/4/2021

Tabla 3.19.: Relación Participa

Se asigna la actividad al grupo donde se ha creado.

idGrupo	idActividad
0	0

Tabla 3.20.: Relación SeOrganizaPara

Se crea una configuración para la actividad.

idConfiguración	Nombre
0	Configuración cocina

Tabla 3.21.: Entidad Configuración

Se asigna el lugar donde se va a hacer.

idActividad	IdLugar
0	0

Tabla 3.22.: Relación SeRealiza

Cambia la disponibilidad del lugar dado que está ahora ocupado.

idLugar	idParámetro	Idvalor
0	9	0

Tabla 3.23.: Relación LugarTieneValorDeParámetro

Se establece la relación entre la actividad y su configuración.

idActividad	IdConfiguración
0	0

Tabla 3.24.: Tiene

idCompetencias	Nombre
0	Comunicación
1	Trabajo en equipo
2	Liderazgo
3	Emprendimiento Social

Tabla 3.25.: Entidad Competencias

### 3. Análisis y modelado

Se le asocia la competencias que se va a satisfacer.

idConfiguración	IdCompetencias
0	0
0	1
0	2

Tabla 3.26.: Relación TieneCompetencias

idTipo	Nombre	EsSubtipoDe
29	Comida Exótica	3
30	Comida étnica	3

Tabla 3.27.: Entidad TipoActividad

Se le asocia el tipo de actividad.

idConfiguración	IdTipo
0	29

Tabla 3.28.: Relación Contiene

Se crea los diversos parámetros de la configuración

idConfiguración	IdParámetro	EsSubtipoDe
0	1	
0	3	
0	4	
0	8	

Tabla 3.29.: Relación ConfiguraciónTieneParámetro  
Se le asignan los valores de los parámetros.

idConfiguración	IdParámetro	Idvalor	EsSubtipoDe
0	1	2	
0	3	5	
0	4	1	
0	8	1	

Tabla 3.30.: Relación ConfiguraciónTieneValorParámetro

## 4. Análisis y diseño del sistema

En este capítulo nos centraremos en la construcción de la especificación de los requisitos, en el diseño de la arquitectura e interfaz de nuestra aplicación.

- El objetivo de éste proyecto es la implementación de una aplicación Android con el objetivo del desarrollo de un sistema para la organización de actividades para la práctica de lenguas extranjeras
- El sistema se implementará en una única plataforma accesible a través de un smartphone con el sistema operativo Android, el cuál tendrá mínimo la versión 5.0 (Lollipop).
- La arquitectura será de cliente-servidor, en la que ambos procesos se llevarán a cabo en el mismo ordenador. El cliente será el usuario que accederá al sistema mediante el cuál hará peticiones al servidor.
- Con el objetivo de empatizar con el usuario, llevaremos a cabo una metodología o disciplina enfocada en el usuario (design thinking) y orientada a la generación de ideas innovadoras para la resolución de problemas detectados dentro de nuestra aplicación[Rei12] .

Se ha decidido que el diseño del sistema de recomendación se hará en un capítulo independiente.

### 4.1. Especificación de los requisitos

La especificación de requisitos recoge una descripción de las funcionalidades, es decir, del comportamiento, del sistema. Los requisitos funcionales definen las funcionalidades del sistema. Los requisitos no funcionales: rendimiento, disponibilidad, seguridad, accesibilidad, usabilidad, estabilidad, portabilidad, costo, operatividad, interoperabilidad, escalabilidad, concurrencia, mantenibilidad e interfaz del sistema[Dec].

#### 4.1.1. Requerimientos funcionales

El sistema cumplirá los siguientes requerimientos sobre el funcionamiento:

**RF1:Registrarse.** El sistema registrará a cualquier usuario que se descargue la aplicación. Por lo tanto, el sistema dará de alta a los usuarios, después de obtener la información necesaria sobre los mismos, con el fin de que éstos puedan acceder a sus servicios. Un usuario registrado será un usuario, que después de completar el proceso de registro, podrá identificarse en el sistema.

**RF2:Introducir Credenciales.** El sistema autenticará los usuarios que previamente hayan sido registrados para que accedan a su sesión. Si la autenticación es correcta, el sistema abrirá la correspondiente sesión. En caso contrario, el sistema volverá a solicitar los datos.

**RF3:Creación del grupo.** El sistema registrará cada grupo creado por el usuario cuando este, de toda la información imprescindible sobre el objetivo del grupo.

#### 4. Análisis y diseño del sistema

**RF4:Unirse a un grupo.** El sistema dejará unirse al usuario a un grupo siempre que este no haya llegado al máximo de usuarios establecido por el creador del grupo.

**RF5:Creación de una actividad.** El sistema dejará crear una actividad siempre que se complete la información necesaria, y que anteriormente tenga creado un grupo .

**RF6:Unirse a una actividad.** El sistema dejará unirse al usuario a la actividad, siempre que este no haya llegado al máximo de usuarios establecido por el creador de la actividad. En este caso, el sistema se encargará de unirlo al grupo si este no está completo.

**RF7:Visualización y de los datos.** El sistema dejará al usuario visualizar sus datos personales del perfil al igual que los datos de los grupos y actividades.

**RF8:Buscar.** El usuario tendrá la posibilidad de buscar en función de serie de parámetros establecidos por el sistema.

**RF10:Cerrar sesión.** Los usuarios podrán abandonar el sistema en cualquier momento.Si el usuario abandona el sistema (o la aplicación se cierra por cualquier razón) sin terminar de definir la información necesaria para un registro, el registro no se creará.

**RF11:Establecer lugar.** El usuario podrá registrar su propiedad privada para que el sistema lo almacene en su base de datos.

**RF12:Gestionar parámetros.** El administrador del sistema podrá añadir, modificar o eliminar los parámetros de la base de datos.

##### Actores involucrados.

Un actor es una entidad externa al sistema que solicita una función al sistema, y por tanto guarda una relación con él. Un actor puede ser tanto un ser humano como un dispositivo electrónico e incluso, otro sistema de software.

Los actores con los que interactuará el sistema son los siguientes:

- **Usuario no registrado:** El actor es un usuario no registrado en el sistema. La razón por la cual un usuario no está registrado es debida a que un usuario es completamente nuevo, nunca se ha registrado en el sistema y por lo tanto, el sistema no contiene información sobre él.
- **Usuario registrado:** El actor es un usuario registrado dentro del sistema, que ha completado el proceso de autenticación y puede acceder a todas las funciones del sistema.
- **Administrador:** El actor es el administrador del sistema. Este actor tiene privilegios en todo el sistema (respetando los datos privados de los usuarios registrados) por lo que puede visualizar y modificar datos de la base de datos.

##### Casos de uso(Diagrama).

Los casos de uso describen el modo en que un actor interactúa con el sistema (descripción de un rol en lenguaje natural). Los diagramas de casos de uso representan los casos de uso de manera gráfica, tal y como se hace en la Figura 4.1, 4.2 y 4.3.

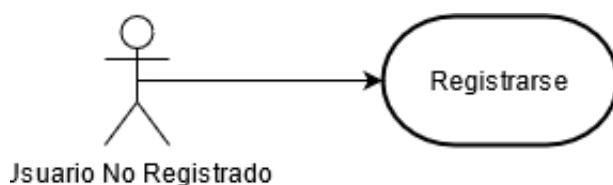


Figura 4.1.: Diagrama de casos de uso para un usuario no registrado.



Figura 4.2.: Diagrama de casos de uso para un usuario registrado.

4. Análisis y diseño del sistema

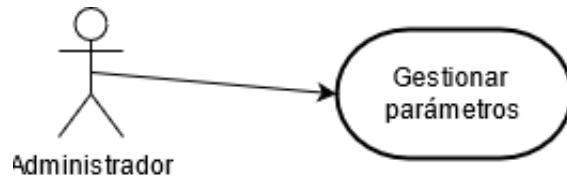


Figura 4.3.: Diagrama de casos de uso para el administrador.

A continuación, los casos de uso representados en las tres figuras anteriores serán descritos detalladamente mediante las siguientes tablas.

<b>Nombre</b>	Registrarse.
<b>Objetivo</b>	Registrar los datos de un nuevo usuario en el sistema.
<b>Actores</b>	Usuario no registrado.
<b>Descripción</b>	El usuario accede a la aplicación para registrarse. El sistema solicita al usuario no registrado datos personales para el registro.
<b>Post-condición</b>	El usuario no registrado pasa a ser un usuario y poder iniciar sesión.

Tabla 4.1.: Tabla caso de uso: Registrarse.

<b>Nombre</b>	Introducir credenciales.
<b>Objetivo</b>	Comprobar si el usuario está registrado en el sistema, para acceder a la aplicación.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario introduce tanto su nombre de usuario como la contraseña. El sistema accederá a la base de datos para comprobar si el usuario está registrado. En el caso afirmativo el usuario accederá al sistema. En el caso contrario el sistema le volverá a pedir los datos.
<b>Post-condición</b>	El usuario registrado pasará a la ventana de inicio de la aplicación.

Tabla 4.2.: Tabla caso de uso: Introducir credenciales.

<b>Nombre</b>	Ver perfil.
<b>Objetivo</b>	El usuario podrá ver sus datos personales en la aplicación.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario desde la ventana principal podrá acceder solo a sus datos personales.
<b>Post-condición</b>	El usuario registrado pasará de ventana de inicio de la aplicación al perfil.

Tabla 4.3.: Tabla caso de uso: Ver perfil.

#### 4.1. Especificación de los requisitos

<b>Nombre</b>	Cerrar sesión.
<b>Objetivo</b>	Cerrar la sesión de un usuario.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario cierra su sesión pulsando el botón de salir.
<b>Post-condición</b>	El usuario registrado pasará de ventana de inicio de la aplicación al inicio de sesión. El usuario podrá volver a iniciar sesión o usar otra cuenta.

Tabla 4.4.: Tabla caso de uso: Cerrar sesión.

<b>Nombre</b>	Establecer un lugar.
<b>Objetivo</b>	El usuario podrá registrar un lugar determinado para su uso posterior.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario pulsará el ícono con forma de casa para acceder.  El usuario llenará un formulario con los datos suficiente para describir el lugar. En el deberá seleccionar el tipo de lugar al que corresponde a través de un desplegable. Tambien tendrá la posibilidad de elegir si en ese momento está disponible o no.
<b>Post-condición</b>	El usuario registrado tras registrar el lugar estos serán almacenado en la base de datos. El usuario volverá a la ventana inicial. Este establecimiento creado será usado posteriormente en la creación de la actividad.

Tabla 4.5.: Tabla caso de uso: Establecer un lugar.

<b>Nombre</b>	Buscar.
<b>Objetivo</b>	El usuario podrá elegir entre varios parámetros para buscar actividades específicas.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario pulsará el ícono con forma de lupa para acceder.  El usuario llenará un formulario seleccionando los parámetros que deseen. El usuario trasselccionar los parámetros tendrán que pulsar el botón de buscar.
<b>Post-condición</b>	Tras hacer la consulta, volverá a la pestaña principal con las actividades filtradas.

Tabla 4.6.: Tabla caso de uso: Buscar.

<b>Nombre</b>	Crear un grupo.
<b>Objetivo</b>	El usuario creará un grupo con una serie de parámetros.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario pulsará el botón con el ícono de grupo. Rellenaremos el formulario que se muestra y estos se registrarán en la base de datos.
<b>Post-condición</b>	Tras pulsar el botón de crear grupo, volveremos a la pestaña inicial donde veremos el grupo creado.

Tabla 4.7.: Crear un grupo.

4. Análisis y diseño del sistema

<b>Nombre</b>	Crear una actividad.
<b>Objetivo</b>	El usuario creará un grupo con una serie de parámetros.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario pulsará el botón con el icono de actividad. A partir de ahí, crearemos un formulario donde se pide una serie de datos y parámetros mínimos. El usuario también deberá elegir el tipo de actividad con la que está relacionado su actividad. Lo mismo pasa con la competencia que satisface. Estos se registrarán en la base de datos.
<b>Pre-condición</b>	El usuario debe anteriormente crear un grupo para poder crear una actividad.
<b>Post-condición</b>	Tras pulsar el botón de crear actividad, volveremos a la pestaña inicial donde veremos la actividad

Tabla 4.8.: Crear una actividad.

<b>Nombre</b>	Formar parte de un grupo.
<b>Objetivo</b>	El usuario se unirá a un grupo.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario en la pestaña principal pulsará el botón para unirse. Se registrará en la base de datos.
<b>Pre-condición</b>	Debe existir al menos un grupo al que unirse. Ese grupo no debe ser creado por el.
<b>Post-condición</b>	Tras pulsar el botón de unirse. El grupo pasará a la pestaña de "mis grupos"

Tabla 4.9.: Tabla caso de uso: Formar parte de un grupo.

<b>Nombre</b>	Formar parte de una actividad.
<b>Objetivo</b>	El usuario se unirá a un grupo.
<b>Actores</b>	Usuario registrado.
<b>Descripción</b>	El usuario en la pestaña principal pulsará el botón para unirse. Se registrará en la base de datos.
<b>Pre-condición</b>	Debe existir al menos una actividad a la que unirse. Esa actividad no debe ser creado por el.
<b>Post-condición</b>	Tras pulsar el botón de unirse. El grupo pasará a la pestaña de "mis actividades"

Tabla 4.10.: Tabla caso de uso: Formar parte de una actividad.

<b>Nombre</b>	Gestionar parámetros.
<b>Objetivo</b>	El administrador accederá a la base de datos para gestionarla.
<b>Actores</b>	Administrador.
<b>Descripción</b>	El administrador pondrá acceder y la base de datos para poder enriquecer los parámetros. Además, pondrá solucionar problemas que le surgen a los usuarios.
<b>Pre-condición</b>	Acceso a la base de datos y servidor.
<b>Post-condición</b>	Dependerá de las acciones del administrador.

Tabla 4.11.: Gestionar parámetros.

#### 4.1.2. Requerimientos no funcionales

##### Interfaz.

En objetivo de nuestra interfaz es que sea lo más sencilla e intuitiva. La interfaz tendrá los módulos dispuestos de manera que visualmente queden bien diferenciados. Se usarán colores agradables a la vista y un tamaño de letra óptimo para su correcta visualización. Estos módulos principales estarán bien diferenciados en diferentes tipos:

1. El componente llamado cards[**Goob**].
2. El componente llamado Text Fileds[**Gooe**].
3. El componente llamado SnackBar[**Good**].
4. El componente llamado buttons[**Gooa**].

Este diseño está basado en un modelo estandar de Android[**Gooc**]. De esta parte se hablará en el capítulo de diseño de la interfaz.

##### Rendimiento.

El tiempo de respuesta del sistema será rápido, tanto para el registro y la autenticación del usuario, como para la introducción y actualización de datos. Este tiempo de respuesta dependerá del servidor donde está alojado la base de datos.

El algoritmo se optimizará de tal manera que se pueda proporcionar una respuesta rápida y fiable al usuario.

##### Fiabilidad, seguridad y estabilidad.

El sistema asegurará fiabilidad en la autenticación de los usuarios, así como la protección de datos. El sistema se recuperará tanto frente a fallos de conexión asegurando que no se pierdan datos del perfil de usuario, como a fallos de sistema haciendo posible su reinicio.

##### Escalabilidad.

El sistema se ha diseñado de tal forma que podemos añadir nuevos módulos y características con bastante facilidad.

#### 4.1.3. Restricciones

El usuario necesitará disponer de los siguientes programas instalados en su ordenador o portátil para poder ejecutar la aplicación:

1. Android Studio 4.2(Gradle 4.2.0).

#### *4. Análisis y diseño del sistema*

2. Dispositivo de emulación Pixel4 API 24.
3. PgAdmin 4 (Version 5.0).
4. PostgreSQL 12.

Además, el sistema contará con las siguientes restricciones:

1. A la hora de crear un usuario, una actividad o un grupo, el usuario deberá de llenar todos los campos para el funcionamiento correcto de la aplicación.
2. Si el usuario no selecciona unos de los campos despegables se escogerá el seleccionado por defecto.

## **4.2. Diseño de la arquitectura**

En este apartado veremos cómo está estructurado el sistema, atendiendo tanto a los componentes como a su relación entre ellos.

### **4.2.1. Esquema general de la arquitectura de la aplicación**

La arquitectura que vamos a usar para nuestra aplicación será una arquitectura Cliente-Servidor. Esta se define como una arquitectura de computación en la que se consigue un procesamiento de la información por medio de procesadores, de tal forma que uno o varios clientes, distribuidos geográficamente o no, solicitan servicios de computación a uno o más servidores[BCC<sup>+</sup>].

Dentro de las modelo, hemos escogido una arquitectura por niveles, que es extensión del modelo cliente/servidor en la que se crea una capa intermedia para la ejecución de la aplicación de gestión de la información

La arquitectura se muestra en la figura 4.4 y se compone de tres capas:

1. Capa de presentación. Contienen procesos cliente simples cuya labor queda relegada a la de interfaz gráfico para la comunicación con el usuario
2. Capa de aplicaciones. Actúan como servidores frente a la capa de presentación y como clientes de la capa de datos. Puede haber varios PCs en esta capa y en la de datos.
3. Capa de datos. Actúan como servidores de datos, conteniendo exclusivamente el código asociado fuertemente a los datos y común a todas las aplicaciones.

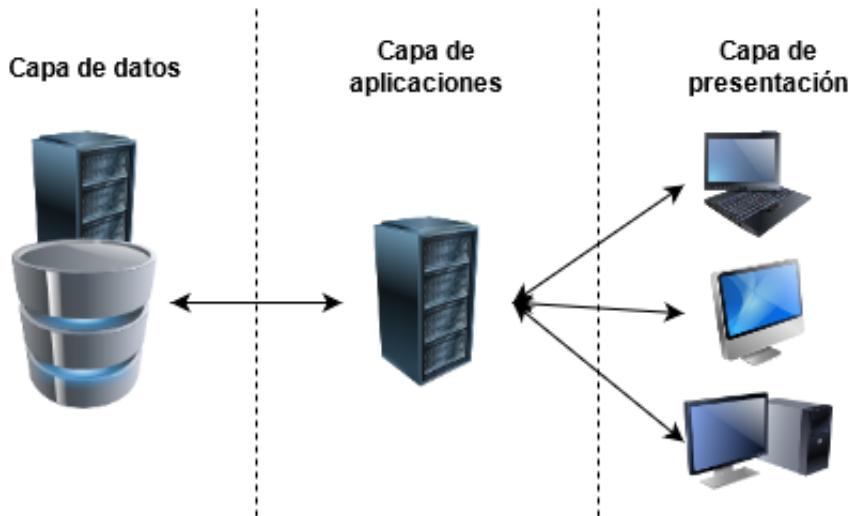


Figura 4.4.: Esquema genérico para una arquitectura por niveles

#### 4.2.2. Diseño técnico de la arquitectura de la aplicación

El diseño técnico de una aplicación proporciona una descripción detallada de los componentes software de la aplicación, así como del modelo de datos.

La aplicación ha sido desarrollada bajo el lenguaje de programación de objetos kotlin, siguiendo una arquitectura cliente-servidor por niveles, con el objetivo de construir una aplicación móvil siguiendo los correspondientes estándares.

El lenguaje de programación bajo el que se ha implementado la aplicación móvil es **kotlin**[dAf] que es un lenguaje de programación estático de código abierto que admite la programación funcional y orientada a objetos. Proporciona una sintaxis y conceptos similares a los de otros lenguajes, como Java y Scala, entre muchos otros. No pretende ser único, sino que se inspira en décadas de desarrollo del lenguaje. Cuenta con variantes que se orientan a la JVM (Kotlin/JVM), JavaScript (Kotlin/JS) y el código nativo (Kotlin/Native).

El programa **Android Studio Arctic Fox**[dAc] que corresponde a la última versión, fue el elegido para implementar la aplicación móvil porque permite construir aplicaciones móviles en el lenguaje de programación kotlin, es multiplataforma y de código abierto, es decir, gratuito; además de por el conocimiento previo sobre el funcionamiento del entorno. También la facilidad que nos da para poder interpretar nuestro modelo de datos.

Pasaremos a explicar los componentes que interviene en cada una de las capas.

1. La implementación de la capa de presentación decidió llevarse a cabo utilizando **XML Android**[dAb]. Al usar vocabulario XML de Android, puedes crear rápidamente diseños de IU y de los elementos de pantalla que contienen, de la misma manera que creas páginas web en HTML, con una serie de elementos anidados. Cada archivo de diseño debe contener exactamente un elemento raíz, que debe ser un objeto View o ViewGroup. Una vez que hayas definido el elemento raíz, puedes agregar widgets u objetos de diseño adicionales como elementos secundarios para crear gradualmente una jerarquía de vistas que defina el diseño. El uso de XML Android nos permite el uso de un gran número de bibliotecas necesarias para poder construir una interfaz competentes. Las librerías usadas estarán descritas en el Anexo A.

#### 4. Análisis y diseño del sistema

2. La conexión que existe entre la interfaz gráfica y activity/Fragment lo conocemos como **view[dAd]** que es cuando compilamos la aplicación, cada archivo XML de diseño se compila en un recurso View.
3. La implementación de la capa de aplicación se decidió utilizarse **activity/Fragment Android[dAa]**. El uso de la clase Activity es un componente clave de una app para Android, y la forma en que se inician y se crean las actividades es una parte fundamental del modelo de aplicación de la plataforma. A diferencia de los paradigmas de programación en los que las apps se inician con un método main(), el sistema Android inicia el código en una instancia de Activity invocando métodos de devolución de llamada específicos que corresponden a etapas específicas de su ciclo de vida. En el caso de la clase Fragment representa un comportamiento o una parte de la interfaz de usuario en una FragmentActivity. Puedes combinar varios fragmentos en una sola actividad para crear una IU multipanel y volver a usar un fragmento en diferentes actividades. Podemos considerarlo como si fuera una subactividad.

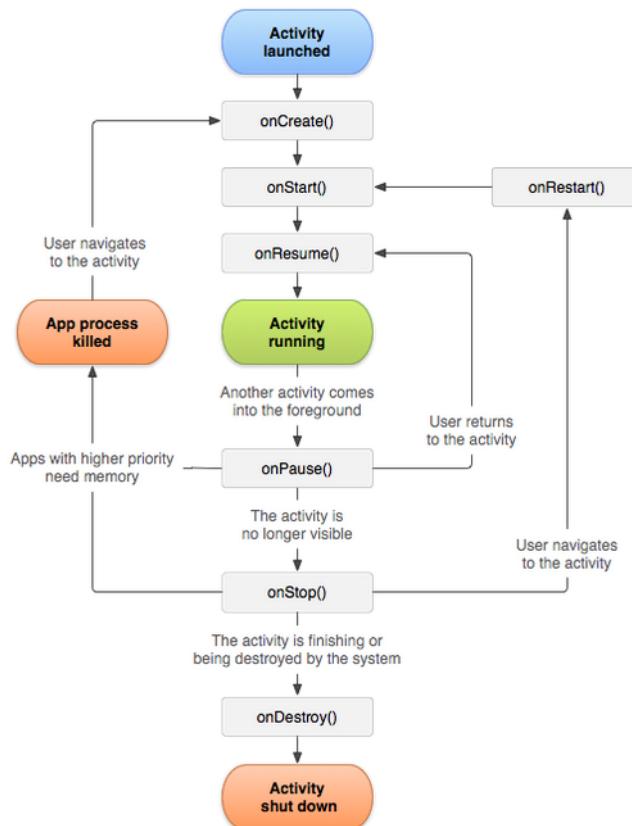


Figura 4.5.: Ilustración simplificada del ciclo de vida de una actividad[dAe].

4. El encargado de la conexión entre la capa de lógica y la capa de datos es **Java Database Connectivity**, más conocido como JDBC [Grob], una interfaz de programación de aplicaciones, es decir, una biblioteca que ofrece un conjunto de funciones y métodos para ser usados por otro software como una capa de abstracción, que permite la

ejecución de operaciones sobre bases de datos para el lenguaje de programación Java. Por lo tanto, JDBC es la biblioteca encargada de la conexión entre la capa de lógica y la capa de datos, es decir, específicamente entre la aplicación web y la base de datos, permitiendo cualquier tipo de consulta de la base de datos, así como creación, modificación o eliminación de datos en ella.

5. Para nuestra ultima capa de datos se ha decidido el uso **PostgreSQL** [Groa] que es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD. El principal razón de su uso de debe a que necesitamos que sea base de datos sql dado nuestro modelo de datos que hemos construido.

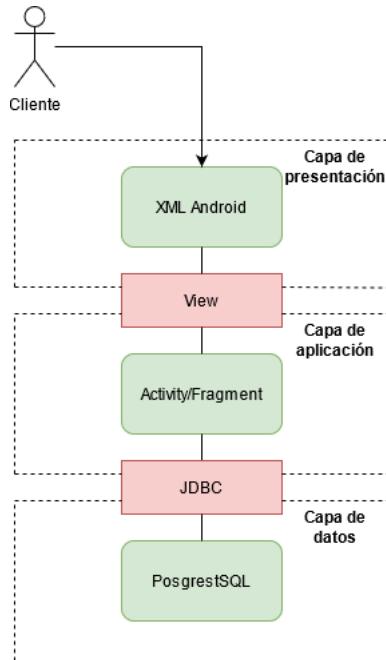


Figura 4.6.: Arquitectura de la aplicación web con las tecnologías empleadas.

#### 4.2.3. Capa de presentación

En éste apartado se detallarán las pantallas de las cuáles está compuesta la capa de presentación, indicando la definición de cada una de ellas, así como las acciones que desempeñan.

En la aplicación encontramos un directorio, perteneciente a nuestro proyecto, llamado res. En el encontramos los ficheros.xml de los que está formada nuestra interfaz.

#### 4. Análisis y diseño del sistema

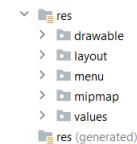


Figura 4.7.: directorio res

1. Los ficheros **drawable** contendrán los iconos de la aplicación.
2. Los ficheros **layout** contienen todos los xml por los que está compuesto la interfaz de la aplicación. En ella podemos encontrar los layout que hacen referencia a las activity, fragment y los items.
3. El fichero **menu** se encuentra el menú de navegación en la parte principal de la aplicación.
4. El fichero **mipmap** podemos encontrar el icono por el cual se lanza la aplicación.
5. Los fichero **values** encontramos los ficheros donde guardamos los valores, los colores y strings por el cual está formado nuestra aplicación.

Ahora detallaremos cada pantalla por las que está formado nuestra aplicación.

<b>Pantalla de registro de datos</b>
<b>Definición:</b> Ésta pantalla permite al usuario introducir los datos requeridos, para completar la etapa del registro.
<b>Acciones:</b>
<b>1.</b> Introducir los datos. <b>2.</b> Acceder al proceso de autentificación.

Tabla 4.12.: activitysign.xml

<b>Pantalla de autenticación</b>
<b>Definición:</b> Esta pantalla permite al usuario introducir sus datos para acceder al sistema. Así como, permitir el acceso al proceso de registro.
<b>Acciones:</b>
<b>1.</b> Acceder al sistema. <b>2.</b> Registrar un nuevo usuario.

Tabla 4.13.: activitylogin.xml

<b>Pantalla de búsqueda de actividades</b>
<b>Definición:</b> Esta pantalla permite al usuario seleccionar diferentes parámetros para encontrar la actividad deseada
<b>Acciones:</b>
<b>1.</b> Seleccionar los parámetros. <b>2.</b> Volver a la pantalla principal.

Tabla 4.14.: activitybusqueda.xml

<b>Pantalla de creación de una actividad.</b>
<b>Definición:</b> Esta pantalla permite al usuario crear una actividad, rellenando todos los campos y seleccionando los parámetros.
<b>Acciones:</b>
1.Seleccionar los parámetros.
2.Rellenar los datos restantes.
3.Volver a la pantalla principal.

Tabla 4.15.: activitycrearactividad.xml

<b>Pantalla de creación de un establecimiento.</b>
<b>Definición:</b> Esta pantalla permite al usuario crear un establecimiento, rellenando todos los campos y seleccionando los parámetros.
<b>Acciones:</b>
1.Seleccionar los parámetros.
2.Rellenar los datos restantes.
3.Volver a la pantalla principal.

Tabla 4.16.: activitycrearestablecimiento.xml

<b>Pantalla de creación de un grupo.</b>
<b>Definición:</b> Esta pantalla permite al usuario crear un grupo, rellenando todos los campos y seleccionando los parámetros.
<b>Acciones:</b>
1.Seleccionar los parámetros.
2.Rellenar los datos restantes.
3.Volver a la pantalla principal.

Tabla 4.17.: activitycreargrupo2.xml

<b>Pantalla principal de la aplicación.</b>
<b>Definición:</b> Pantalla principal de nuestro sistema donde podemos ver todos los elementos del sistema.
<b>Acciones:</b>
1.Acceder para ver el menú de la aplicación.
2.Acceder para crear un grupo, actividad, establecimiento, ver el perfil, etc.
3.Acceder para hacer una búsqueda de actividades.

Tabla 4.18.: activitymain.xml

4. Análisis y diseño del sistema

<b>Pantalla del menú de los grupos del usuario.</b>
<b>Definición:</b> En esta pantalla el usuario puede ver todas las grupos en las que se encuentra el usuario.
<b>Acciones:</b>
<b>1.Ver los grupos a los que pertenece el usuario.</b>

Tabla 4.22.: fragmentmygroup.xml

<b>Pantalla del perfil del usuario.</b>
<b>Definición:</b> En esta pantalla el usuario puede ver sus datos personales.
<b>Acciones:</b>
<b>1.Ver los datos personales.</b>
<b>2.Volver al pantalla principal.</b>

Tabla 4.19.: activityperfil.xml

<b>Pantalla del menú de las actividades.</b>
<b>Definición:</b> En esta pantalla el usuario puede ver todas las actividades que hay en el sistema.
<b>Acciones:</b>
<b>1.Ver las actividades.</b>
<b>2.Unirse a una actividad.</b>

Tabla 4.20.: fragmentactivity.xml

<b>Pantalla del menú de las actividades del usuario.</b>
<b>Definición:</b> En esta pantalla el usuario puede ver todas las actividades en las que participa el usuario.
<b>Acciones:</b>
<b>1.Ver las actividades a las que pertenece.</b>

Tabla 4.21.: fragmentmyactivity.xml

<b>Pantalla del menú de los grupos.</b>
<b>Definición:</b> En esta pantalla el usuario puede ver todas las grupos que contiene el sistema.
<b>Acciones:</b>
<b>1.Ver los grupos.</b>
<b>2.Unirse a un grupo.</b>

Tabla 4.23.: fragmentgroup.xml

<b>Lista dinámica de las actividades y grupos.</b>
<b>Definición:</b> Esta pantalla formará parte de una lista dinámica que se muestran en las pantallas iniciales de la aplicación
<b>xml pertenecientes:</b>
1.itemactivity.xml
2.itemgroup.xml
3.itemmygroup.xml
4.itemmyactivity.xml

Tabla 4.24.: Listas dinámicas.

<b>Barra de navegación de la aplicación.</b>
<b>Definición:</b> En esta navegación el usuario podrá cambiar entre las diferentes pantalla de las actividades y grupos.
<b>Acciones:</b>
1.Cambiar de pantalla.

Tabla 4.25.: bottomnavmenu.xml

#### 4.2.4. Capa de aplicaciones

En éste apartado se detallarán, mediante las siguientes tablas, las clases de datos, Activity/-Fragment de las cuáles está compuesta la capa de lógica, indicando la definición de cada una de ellas, así como las acciones que desempeñan. A las Activity y los fragment se le añadirá el diagrama de secuencia.

Comenzaremos definiendo las data class que hemos definido. Una data class no es más que una clase que sólo contiene estado y no realiza ninguna operación. Por lo que no tiene mucha relevancia ya que no realiza ninguna operación.

<b>Clase dataClass</b>
<b>Definición:</b> Esta clase permite contener el estado de todos nuestros objetos en nuestro sistema.
<b>Clases pertenecientes:</b>
1.Actividad.kt
2.Clasemisgrupos.kt
3.Grupos.kt
4.GruposParámetros.kt
5.Lugar.kt
6.LugarParametros.kat
7.Usuario.kt

Tabla 4.26.: Data Class del sistema.

Ahora pasaremos a definir las Activity.

4. Análisis y diseño del sistema

Clase ActividadAdaptador
<b>Definición:</b> Esta clase permite permitir al usuario poder unirse a las actividades que se muestran en la lista que crea. Además, muestra la lista de actividades.
<b>Acciones:</b>
<b>1.</b> Mostrar las actividades. <b>2.</b> Unirse a una actividad.

Tabla 4.27.: ActividadAdaptador.kt

Clase Busqueda
<b>Definición:</b> Esta clase permite permitir al usuario poder hacer una búsqueda de las actividades a partir de una serie de parámetros.
<b>Acciones:</b>
<b>1.</b> Seleccionar el modo de búsqueda. <b>2.</b> Seleccionar la categoría. <b>3.</b> Seleccionar el ranking. <b>4.</b> Hacer la búsqueda.

Tabla 4.28.: Busqueda.kt

Clase Conect
<b>Definición:</b> Esta clase permite a la aplicación a través de la API de JDBC conectarse conectarse a la base de datos alojada en la nube. Además encontraremos todas las sentencias SQL para hacer las peticiones que necesita nuestro sistema.
<b>Acciones:</b>
<b>1.</b> Configurar la conexión. <b>2.</b> Conectarse a la base de datos. <b>3.</b> Hacer peticiones SQL.

Tabla 4.29.: conect.kt

Clase Crear Actividad
<b>Definición:</b> Esta clase permite al usuario crear una actividad rellenando los datos que pide el sistema.
<b>Acciones:</b>
<b>1.</b> Seleccionar los parámetros. <b>2.</b> Rellenar los datos que quedan. <b>3.</b> Crear actividad.

Tabla 4.30.: CrearActividad.kt

<b>Clase Crear Establecimiento</b>
<b>Definición:</b> Esta clase permite al usuario crear un establecimiento rellenando los datos que pide el sistema.
<b>Acciones:</b>
1.Seleccionar los parámetros. 2.Rellenar los datos que quedan. 3.Crear establecimiento.

Tabla 4.31.: CrearActivityEstablecimiento.kt

<b>Clase CrearGrupo</b>
<b>Definición:</b> Esta clase permite al usuario crear un grupo rellenando los datos que pide el sistema.
<b>Acciones:</b>
1.Seleccionar los parámetros. 2.Rellenar los datos que quedan. 3.Crear grupo.

Tabla 4.32.: CrearGrupoActivity2.kt

<b>Clase GrupoAdaptador</b>
<b>Definición:</b> Esta clase permite al usuario poder unirse a un grupo. Además permite mostrar una lista de todos los grupos disponibles.
<b>Acciones:</b>
1.Mostrar los grupos. 2.Unirse a un grupo.

Tabla 4.33.: GrupoAdaptador.kt

<b>Clase Login</b>
<b>Definición:</b> Esta clase permite a un usuario poder autenticarse para poder acceder al sistema. También permite registrarse.
<b>Acciones:</b>
1.Escribir sus credenciales. 2.Autenticarse. 3.Registrarse.

Tabla 4.34.: Login.kt

4. Análisis y diseño del sistema

Clase MainActivity
<b>Definición:</b> Esta clase se encarga de la vista principal de nuestra aplicación.
<b>En ella el usuario puede ver las actividades y grupos, crearlas o poder ver su perfil.</b>
<b>Acciones:</b>
<b>1.Seleccionar la barra de navegación.</b>
<b>2.Crear actividades y grupos.</b>
<b>3.Ver su perfil.</b>
<b>4.Crear establecimientos.</b>
<b>5.Hacer busquedas.</b>

Tabla 4.35.: MainActivity.kt

Clase MiActividadAdaptador
<b>Definición:</b> Esta clase se encarga de mostrar las actividades en los que el usuario se ha unido.
<b>La diferencia con el otro adaptador es que muestra toda la información de la actividad.</b>
<b>Acciones:</b>
<b>1.Mostrar las actividades del usuario.</b>

Tabla 4.36.: MiActividadAdaptador.kt

Clase MiGrupoAdaptador
<b>Definición:</b> Esta clase se encarga de mostrar los grupo en los que el usuario se ha unido.
<b>La diferencia con el otro adaptador es que muestra toda la información del grupo.</b>
<b>Acciones:</b>
<b>1.Mostrar los grupos del usuario.</b>

Tabla 4.37.: MiGrupoAdaptador.kt

Clase Perfil
<b>Definición:</b> Esta clase se encarga de mostrar los datos personales del usuario.
<b>Acciones:</b>
<b>1.Mostrar los datos del usuario.</b>

Tabla 4.38.: PerfilActivity.kt

Clase Sign
<b>Definición:</b> Esta clase se encarga de registrar a nuevo usuario.
<b>Acciones:</b>
<b>1.Rellenar los datos personales.</b>
<b>2.Registrarse.</b>

Tabla 4.39.: Sign.kt

Por último, definiremos los fragment.

<b>Clase ActivityFragment</b>
<b>Definición:</b> Esta clase se encarga de hacer la petición a la clase conect, para que esta haga la petición a la base de datos y le devuelva las actividades. Por último, esta le envía las actividades al adaptador.
<b>Acciones:</b>
1.Petición a la API 2.Le envía la información al adaptador.

Tabla 4.40.: ActivityFragment.kt

<b>Clase MyActivityFragment</b>
<b>Definición:</b> Esta clase se encarga de hacer la petición a la clase conect, para que esta haga la petición a la base de datos y le devuelva las actividades, con toda la información adicional. Por último, esta le envía las actividades al adaptador.
<b>Acciones:</b>
1.Petición a la API 2.Le envía la información al adaptador.

Tabla 4.41.: MyActivityFragment.kt

<b>Clase GroupFragment</b>
<b>Definición:</b> Esta clase se encarga de hacer la petición a la clase conect, para que esta haga la petición a la base de datos y le devuelva los grupos. Por último, esta le envía los grupos al adaptador.
<b>Acciones:</b>
1.Petición a la API 2.Le envía la información al adaptador.

Tabla 4.42.: GroupFragment.kt

<b>Clase MyGroupFragment</b>
<b>Definición:</b> Esta clase se encarga de hacer la petición a la clase conect, para que esta haga la petición a la base de datos y le devuelva los grupos con información adicional. Por último, esta le envía los grupos al adaptador.
<b>Acciones:</b>
1.Petición a la API 2.Le envía la información al adaptador.

Tabla 4.43.: MyGroupFragment.kt

#### 4.2.5. Capa de datos

La base de datos de la aplicación contiene la información necesaria para que la aplicación pueda hacer uso de ella y así proporcionar su servicio de manera correcta. Como ya explicamos en el capítulo anterior de modelado las tablas que teníamos y mostramos sus respectivos diagramas, pasaremos a la descripción de cada tabla ahora en el sistema.

#### 4. Análisis y diseño del sistema

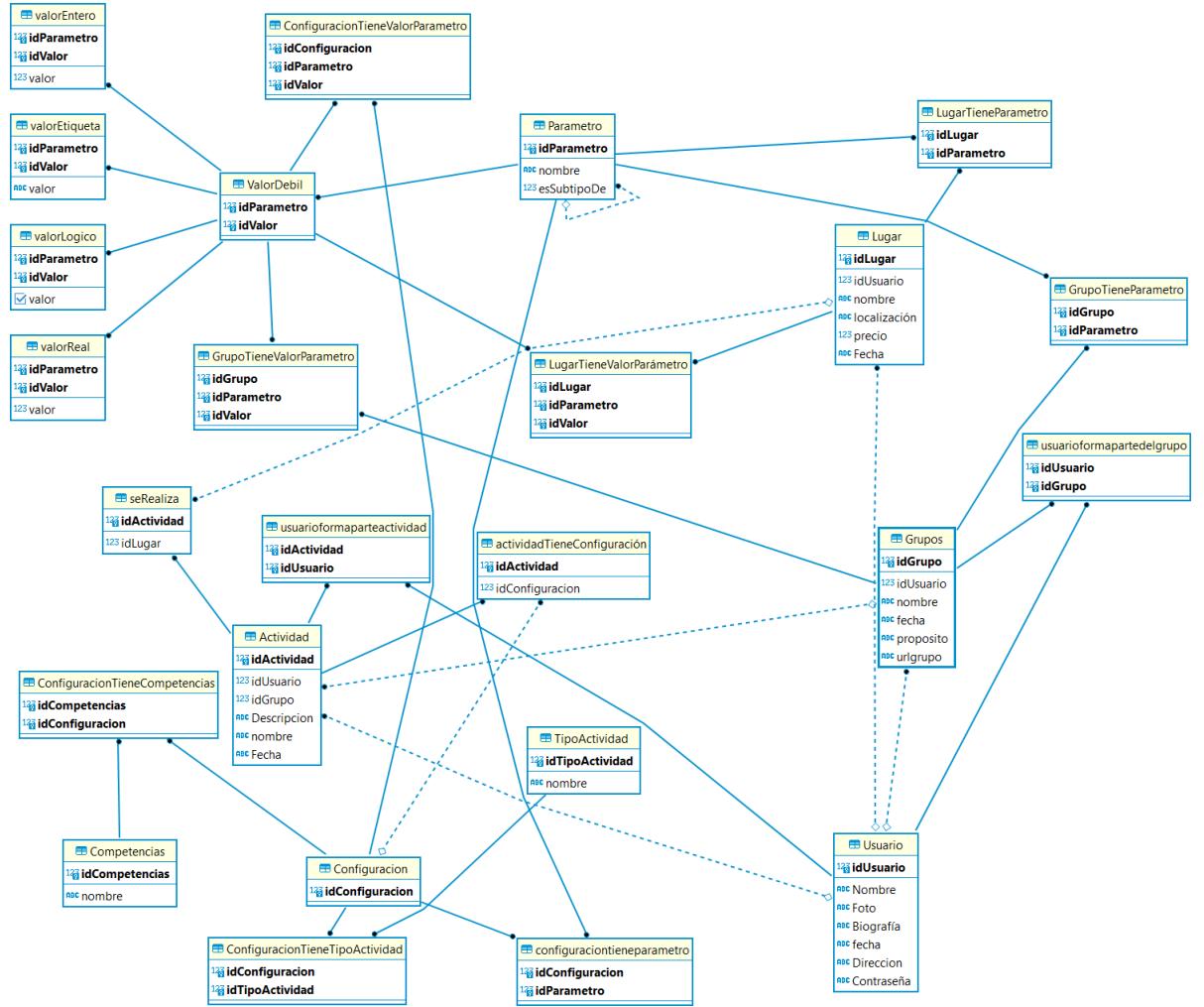


Figura 4.8.: Base de datos en postgresSql generado por DBeaver.

Tal y como se puede apreciar en el diagrama, la base de datos está formada por un total de 25 tablas, comenzaremos describiendo las entidades y luego las relaciones.

1. La tabla **Actividad** almacena la información correspondiente a las actividades creados por los usuarios, es decir, la información necesaria para que el usuario conozca los datos básicos de la actividad.

Número	Columnas	Tipo	Restricciones	Descripción
1	idActividad	Int	PRIMARY KEY	Id de la actividad.
2	idUsuario	Int	FOREIGN KEY	Id del usuario propietario.
3	idGrupo	Int	FOREIGN KEY	Id del grupo perteneciente.
4	Descripcion	character varying		Descripción de la actividad.
5	nombre	character varying		Nombre de la actividad.
6	Fecha	character varying		Fecha de la actividad.

Tabla 4.44.: Información de la tabla Actividad.

Para evitar alargar demasiado este capítulo, además de que en el capítulo 3 se explicó, el resto de las tablas podrán ser consultadas en el anexo A.1.

### 4.3. Diseño de la seguridad

Con el fin de evitar ataques y brechas en la seguridad, se decide cifrar datos de usuarios como las contraseñas con md5.

Con esto se consigue que, aunque alguien consiguiera acceso a la base de datos y extrajera datos de los usuarios, dichos datos estarían cifrados, teniendo así que recurrir al uso de una fuerza bruta muy costosa.

Por ahora, la seguridad que incluye la aplicación es muy básica, se deberá ampliar para el producto final, siendo recomendable ampliar al modelo de seguridad OWASP, marco de requisitos de seguridad necesarios para diseñar, desarrollar y probar aplicaciones móviles seguras en iOS y Android. Todo para evitar futuros ataques.

### 4.4. Diseño de la interfaz

En esta parte, nos centraremos principalmente en el diseño, la jerarquía de información, y la aparición de elementos de interfaz en pantallas de nuestra aplicación móvil.

Por otra parte, también incluiremos las ideas creativas, de economía colaborativa y negocio, con el fin de satisfacer las demandas buscadas por los usuarios.

Para todo esto, seguiremos una metodología de desarrollo en ese proceso del diseño buscando la competencia actual para poder buscar sus puntos débiles y fuertes, poder clasificar esa información, organización de la información que se va a presentar en la aplicación, primeros bocetos que nos ayudará a comprender si es buena idea colocar esa información en tal sitio, un diseño más avanzado usando herramientas más profesionales, y a todo esto hacerle una evaluación.

Para el diseño de la interfaz nos basaremos en un diseño anterior con las modificaciones pertinentes y necesarias. Estas modificaciones están relacionadas con lo aprendido sobre actividades en el capítulo 2. Para adaptarlo nos hemos basado en un modelo que se nos proporcionó [Rei12]

#### 4.4.1. Introducción

¿ Algunas vez has visto grupos de personas practicando deportes o aficiones que a ti también te gustaría hacer, pero no conoces a nadie?

#### 4. Análisis y diseño del sistema

Esta fue la pregunta que nos planteamos y fue el incentivo que nos llevo a desarrollar este proyecto.

Nuestro principal objetivo era fomentar las actividades en grupo poniendo en contacto a personas con los mismo gustos. Por ello hemos llevado a cabo el desarrollo de una aplicación que pueda cubrir estas necesidades.

##### 4.4.2. Competitive Analysis

Es obvio que existen decenas de aplicaciones relacionadas con las redes sociales, búsqueda de experiencias con personas de otros lugares o compartir aficiones. Por ello, es recomendable buscar estas aplicaciones y encontrar sus puntos fuertes.

Entre las aplicaciones y páginas web que hemos visto, las que más se adaptan mejor a una búsqueda de actividades y experiencias relacionadas con los idiomas encontramos estas [yIMP2oa].

Para ver las principales características las resumiremos en un cuadro comparativo.

EMPRESAS	Meetup-Language	Verbling	Busuu	Tandem	Hello Talk	Lingvu	Hi uTandem
Gratis	SI	SI	SI	SI	SI	SI	SI
Opción de pago	SI	NO	SI	NO	NO	NO	NO
Más de un idiomas	SI	SI	SI	SI	SI	NO	SI
web	SI	SI	SI	SI	SI	NO	NO
Android	SI	NO	SI	SI	SI	NO	SI
iOS	SI	NO	SI	SI	NO	NO	SI
Proximidad de interlocutores	SI	NO	NO	NO	NO	NO	SI
Videoconferencia	NO	SI	SI	SI	SI	SI	SI
Chat	SI	SI	SI	SI	SI	SI	SI
Herramientas lingüísticas	NO	SI	SI	SI	SI	NO	SI

Figura 4.9.: Tabla comparativa de empresas

Entre las aplicaciones y páginas web que hemos visto las que más se adaptan mejor a una experiencia de ocio son Meetup-Language y Hi uTandem. Entre estas dos se ha elegido Meetup-Language dado que está disponible en Android y no muestra problemas "graves".

##### 4.4.3. Feedback Capture Grid

Hemos analizado a fondo la página Meetup-Language, rellenando una Usability Review donde hemos detectado características importantes que nos pueden ayudar en la experiencia de usuario de la aplicación [yIMP2ob].

Como conclusión al uso de la página Meetup-Language, creamos este diagrama con los aspectos más destacados:

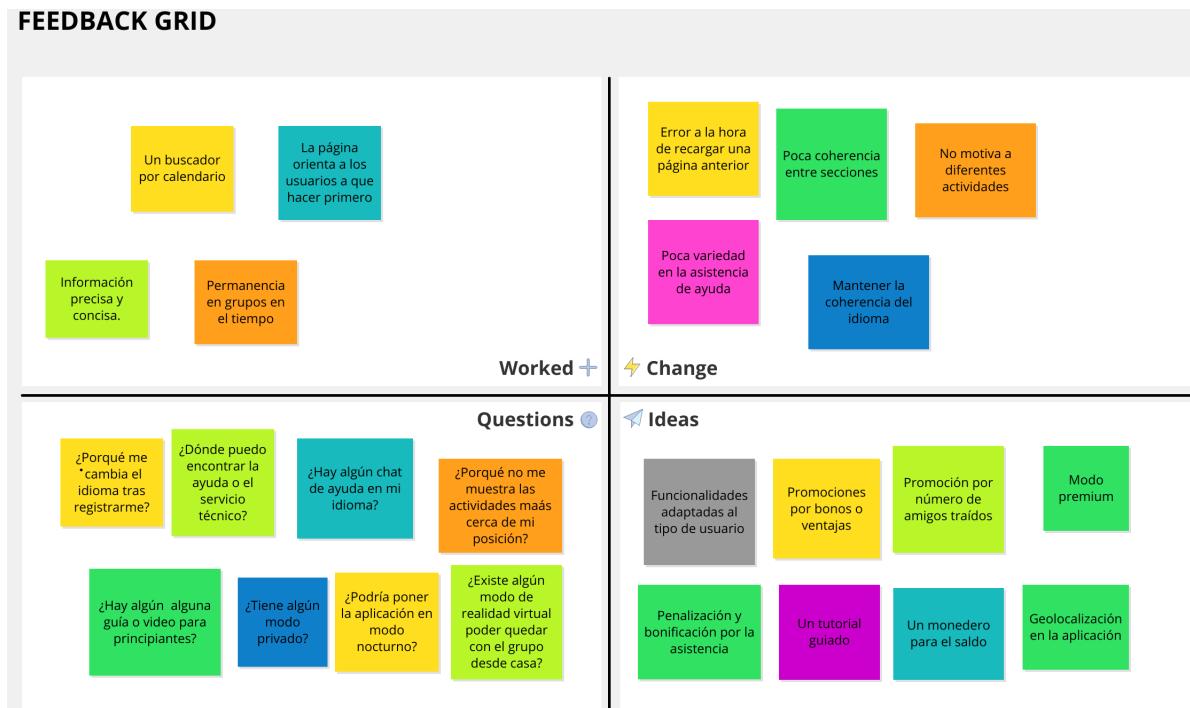


Figura 4.10.: Malla receptora de información

De esta aplicación podemos aprender bastantes cosas:

- No hace llamativa ni motiva las actividades ni eventos.
- Se basa en un conjunto de grupos que hacen actividades, pero no aprovecha el poder de la geolocalización.
- Deberá visualizar información sobre los grupos, y cada uno de los eventos que hacen.
- Cualquier persona podrá ver los eventos y unirse a ellos.
- Al haber posibilidad de que los eventos sean de pago, habría que permitir pasar por una plataforma de pago sencilla y fácil de usar.
- Una vez unido al evento este usuario tendría que estar al pendiente de posibles comunicaciones e incidencias, por lo que la app tendría que tener un sistema de notificaciones y la posibilidad de contactar con quien organiza el evento para cualquier cosa que tenga que ver con el grupo o el evento.
- Una vez finalice el evento el usuario podría valorarlo y unirse al grupo que lo ha hecho por si quiere asistir a más eventos.

#### 4.4.4. Tasks and Sitemap

Entre las dos posibilidades nos hemos decidido por hacer la matriz de tareas para analizar los usuarios que van a usar la aplicación y las tareas principales para poder analizar

#### 4. Análisis y diseño del sistema

la frecuencia de uso de las acciones y así poder situar las funcionalidades en la aplicación. Para todo esto se hará un ranking de estas funcionalidades pero también viendo otros aspectos ya que habrá algunas funcionalidades que solo usará uno de los tantos tipos de usuarios que hay. En este caso se ha modificado pero este se podría adaptar completamente en futuros proyectos.

GRUPOS DE USUARIOS	Usuarios no registrados	Usuario	Ranking
Buscar	NO	Alto	12
Formar parte de una actividad	NO	Alto	6
Formar parte de un grupo	NO	Medio-Alto	5
Crear una actividad	NO	Medio	5
Crear un grupo	NO	Medio	5
Borrar una actividad	NO	Medio-Bajo	4,5
Borrar un grupo	NO	Medio-Bajo	4
Dejar un grupo	NO	Bajo	3
Dejar una actividad	NO	Bajo	2
Iniciar sesión	NO	bajo	2
Introducir credenciales	Bajo	No	1
Registrarse	Bajo	No	1

Tabla 4.45.: Matriz De Tareas y Usuarios.

En cuestión de la jerarquía, este sería el esquema principal del sitemap.

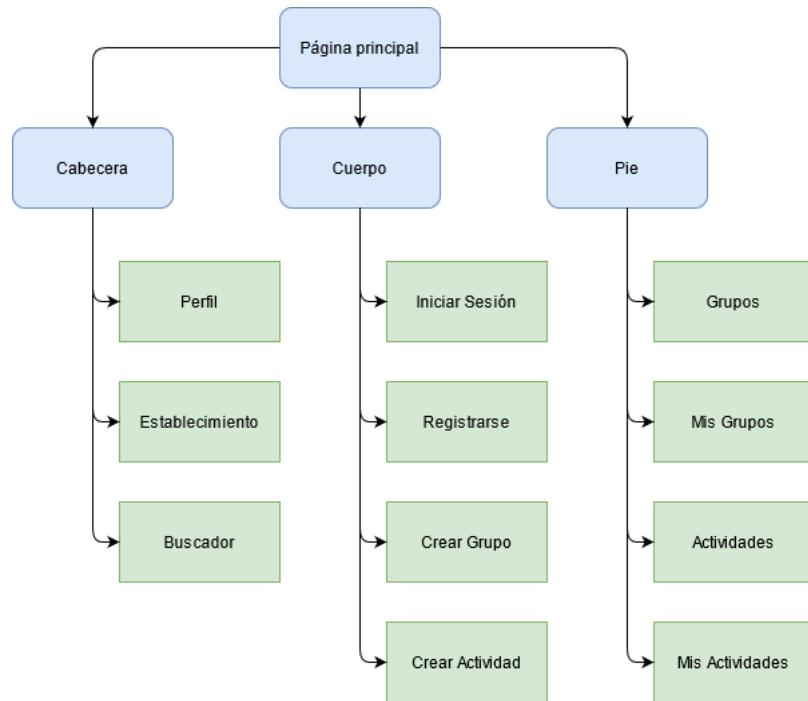


Figura 4.11.: Jerarquía principal.

#### 4.4.5. Labelling

En este apartado describiremos que significa cada botón de acción para poder facilitar la comprensión de los bocetos.

Término	Icono	Significado
Perfil		Hace referencia a los datos del usuario. Nombre usuario, foto de perfil, ciudad y biografía.
Establecimiento		Se accede al apartado para poder crear un establecimiento, punto de reunión o lugar.
Recargar estado		Sincronizar los nuevos estados de las actividades y grupos.
Buscador		Se accede a un apartado de búsqueda por donde se le proporcionará lo solicitado en función de los parámetros seleccionados.
Visualización o creación de grupo		Se crea un grupo donde se describirá de forma breve su propósito, los idiomas que se van a usar, tamaño, etc. También hace referencia a la visualización de los grupos.
Visualización o creación de actividades		Se crea la actividad de un grupo donde se dará una pequeña explicación de las competencias, tipos, fotos, lugar del establecimiento, fecha y hora y el precio de la asistencia. También hace referencia a la visualización de actividades.
Mis grupos		Se visualizan los grupos a los que se ha unido el usuario.
Mis actividades		Se visualizan las actividades a los que se ha unido el usuario.
Unirse a un grupo		El usuario podrá unirse a un grupo mientras existan plazas.
Unirse a una actividad		El usuario podrá unirse a una actividad mientras existan plazas.
Anular actividad o grupo		Opción (aún no disponible) si ya no le interesa asistir al evento o a un grupo al usuario.

Tabla 4.46.: Labelling de la aplicación.

#### 4.4.6. Wireframes

Para esta parte hemos desarrollado una serie de bocetos sobre las distintas partes de la interfaz. Para tener una mayor organización y podamos entender el flujo que sigue se hará un diagrama de flujo de los bocetos. Esta aplicación se ha diseñado de tal forma que se pueda llevar a cabo su implementación tanto en android o ios.

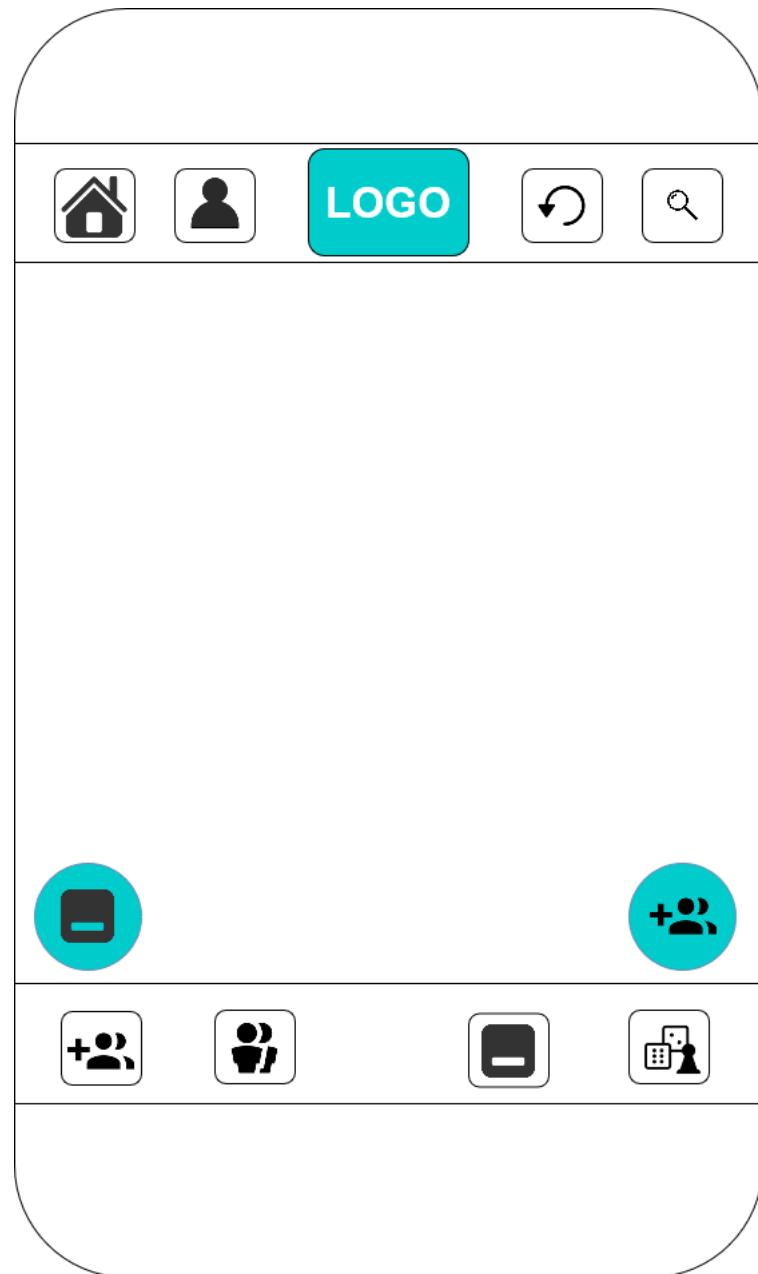


Figura 4.12.: Vista principal de la aplicación

Para no alargar mucho el capítulo, el resto de los bocetos podrán ser consultadas en el anexo A.3.

#### 4.4.7. Logotipo

El logotipo final para este proyecto se ha elegido el siguiente:



Figura 4.13.: Logotipo de la aplicación.

Para ello buscamos información sobre consejos y páginas para su creación. En esta última nos dio la idea de añadir un eslogan aunque no se ha llevado a cabo. El resto de logotipos que también se diseñó como posibles candidatos los encontramos [aquí](#).

#### 4.4.8. Style Guide

En el caso de la tipografía nos basaremos en el sistema de diseño [Material Design](#). Estos tamaños y estilos de texto de Material Design se desarrollaron para equilibrar la densidad del contenido y la comodidad de lectura en condiciones de uso típicas.

Para este proyecto se hizo uso de algunos de estos estilos.

1. Headline 1
2. Caption
3. Overline
4. Subtitle 1
5. Subtitle 2
6. Body 1
7. Body 2
8. Button

Para la elección de los colores se hizo uso también de [Material Design](#). A través de este enlace podrá ver los colores elegidos. Se eligieron estos por el color del logotipo.

#### 4. Análisis y diseño del sistema

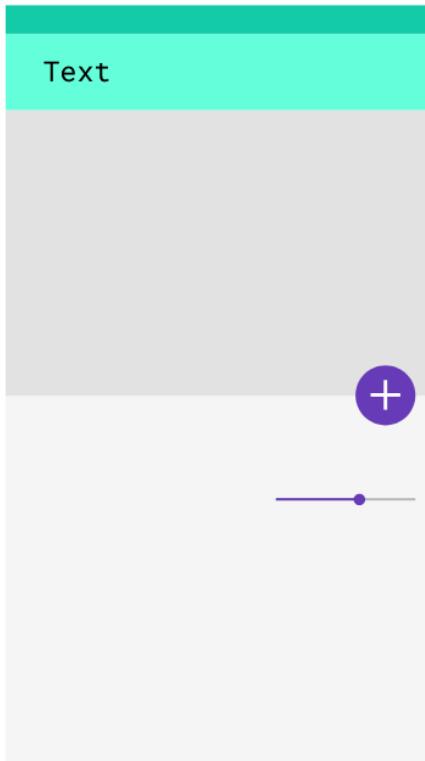


Figura 4.14.: Ejemplo 1

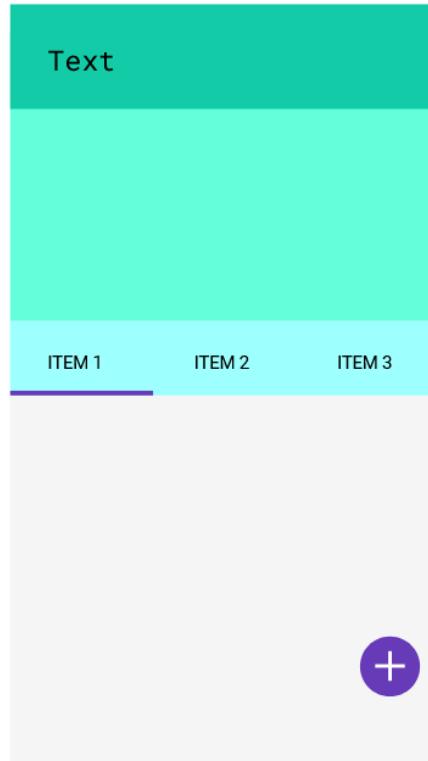


Figura 4.15.: Ejemplo 2

#### 4.4.9. Guidelines

Se ha investigado acerca del sistema de diseño Material Design, descrito en [material.io](#), para su posible implementación en la aplicación.

Se trata de un lenguaje de diseño que destaca por su estudio de los materiales, cómo interactúan entre ellos y cómo comparten un espacio físico y pueden ser apilados, proyectar sombras y presentar profundidad.

El lenguaje Material Design es fácil de implementar, y presenta mucha flexibilidad. Además, es idóneo para aplicaciones móviles y web, y permite expresar la nueva imagen de marca con mayor personalidad, integrándola en el diseño.

Una de las principales características de Material Design es cómo todos los elementos de la interfaz de usuario son definidos como sólidos. Algunas de sus propiedades son las siguientes:

1. No puede haber dos elementos al mismo nivel ocupando el mismo espacio.
2. Los elementos pueden apilarse y colocarse en diferentes capas.
3. Pueden dividirse y recombinarse.
4. Pueden cambiar de opacidad, ya sea uniformemente o de forma parcial.
5. Pueden ampliarse y reducirse únicamente en el plano elementos contenidos, nunca doblarse o extruirse en profundidad.

#### 4.4.10. Maqueta final.

Para la maqueta final de nuestro prototipo se pensó inicialmente en usarse Adobe XD, pero dado la falta de tiempo se empezó a implementar directamente en Android Studio. Observaremos que se ha intentado ajustar a nuestro diseño inicial, pero con algunos diferencias significativas. Estos serían los resultados finales:

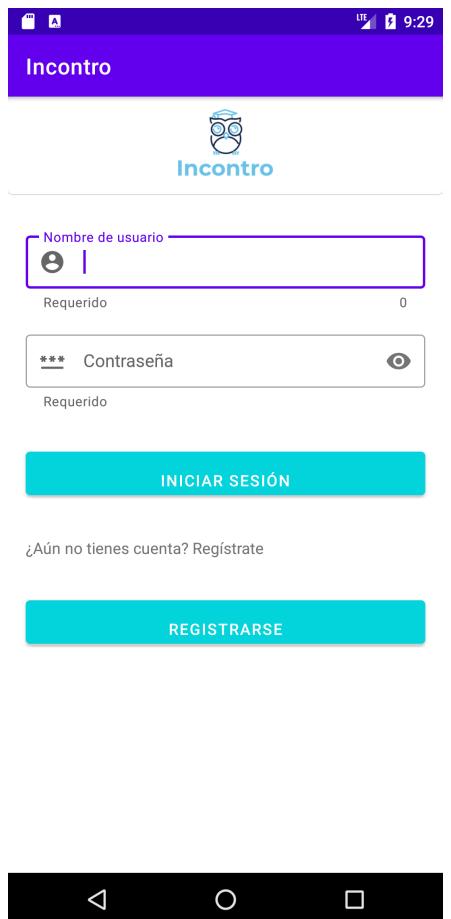


Figura 4.16.: Inicio de sesión.

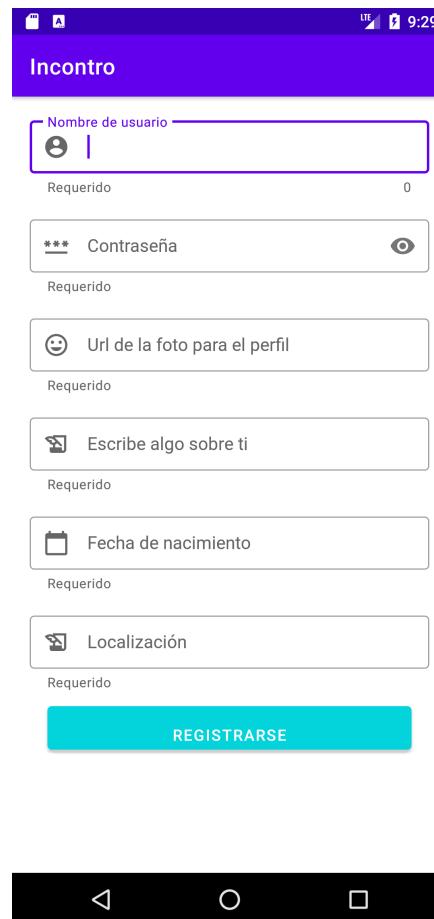


Figura 4.17.: Registrarse.

Para no alargar mucho el capítulo, el resto de los bocetos podrán ser consultadas en el anexo A.3, como hicimos con el diseño inicial.



## 5. Diseño del sistema de recomendación

Uno de los puntos fuertes de nuestro sistema, era la posibilidad de que el usuario pudiera buscar las actividades deseadas a través de un buscador. Siguiendo nuestro modelo de arquitectura elegida y en la filosofía de nuestra aplicación, nuestra base de datos que tenemos en la nube se encargaría de hacer las búsquedas, liberando de la aplicación de cualquier carga de ejecución. Esto significa que nuestro diseño construiría una consulta compleja en SQL, donde el usuario elige que quiere buscar, de que manera quiere se le muestre y si le interesa ver todos los datos que ha escogido, recogiendo todos los datos, ó una interrelación de lo escogido. Otro aspecto importante de la necesidad del buscador es la falta de información que se muestra, antes de que te unas a ella, por lo que el usuario puede estar seguro de que cuando haga una búsqueda tenga esas características esas actividades.

La construcción de estas sentencias complejas pasa por varias fases, dado que los procesos de búsqueda de la categoría y el criterio de búsqueda son procesos aislados entre ellos:

1. En función de si se escoge una o más categorías, construiremos una serie de sentencias unidas que buscarán si existen esas actividades con esas características.
2. Determinar la consulta que resuelve el ranking, a estas se le asigna un valor en función del ranking que se busca.
3. Juntamos ambos a través de una reunión natural y en función del ranking escogido se ordenará de menor a mayor o de mayor a menor.

Por ello nuestro diseño del sistema lo hemos dividido en cuatro partes.

### 5.1. Categorías

En nuestro primer apartado, tenemos como finalidad el objetivo de que el usuario elija que quiere buscar. Para ello hemos seleccionado las características que dan más carácter a nuestras actividades:

- **Idioma.** Es interesante que el usuario pueda ver actividades en función del idioma, ya que es uno de los aspectos más importantes de nuestro sistema.
- **Tamaño de la actividad.** Al usuario le puede interesar actividades en pareja o donde se formen grupos grandes.
- **Fecha.** El usuario puede estar interesado en participar en actividades en un día concreto.
- **Duración de la actividad.** Puede darse el caso que algunas actividades no se satisfagan en un solo día y que se haga a lo largo de una semana, y que al usuario le interese solo las que se complete en un día.

## 5. Diseño del sistema de recomendación

- **Lugar de la actividad.** Al usuario le puede interesar hacer actividades solo en la montaña o en la playa.
- **Tipo de actividad.** Al usuario le puede interesar tipos de actividades en concreto como, por ejemplo, proyectos o gastronomía.
- **Competencias.** Al usuario le pueden interesar satisfacer alguna competencia en concreto.

## 5.2. Consulta simple o compuesta

El principal objetivo es dar la libertad al usuario de que pueda buscar actividades muy específicas o buscar el mayor número de actividades con las opciones seleccionadas. Además de que es una forma de unión entre las categorías, construyendo una sentencia más compleja que pueda satisfacer al usuario.

En este caso podemos encontrar dos tipos. La unión entre las categorías donde el usuario pueda encontrar todas las actividades que tenga alguna relación con la categoría seleccionada, conjuntiva o hacer la intersección buscando una actividad específica, disyuntiva.

## 5.3. Criterio de búsqueda

En esta última parte veremos que tras combinar las dos primera partes, podemos llegar más lejos aun aplicando lo que llamaríamos un "Ranking". Esto significa que con las actividades que hemos obtenido de las dos primeras partes, podríamos asignarle un valor y ordenarlo en función de un criterio. Por ello vamos a diseñar varios criterios.

- **La actividad con más usuarios inscritos en ella.** En este criterio obtendríamos en orden creciente de la actividad con el mayor número de personas registradas en ella. Este sería nuestro esquema de navegación:

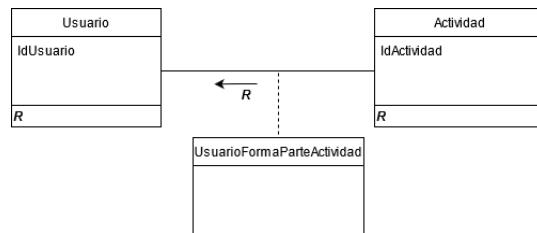


Figura 5.1.: Esquema de navegación.

- **La actividad con más usuarios inscritos en el grupo que pertenece.** En este criterio obtendríamos en orden creciente, el mayor número de personas registradas en el grupo al que pertenece la actividad. Este sería nuestro esquema de navegación:

#### 5.4. Incorporación de nuevos ranking

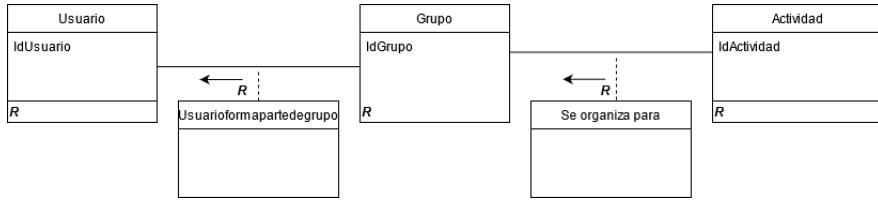


Figura 5.2.: Esquema de navegación.

- **La actividad que satisface el mayor número de competencias.** En este criterio obtendríamos en orden creciente de la actividad con el mayor número de competencias que satisface. Este sería nuestro esquema de navegación:

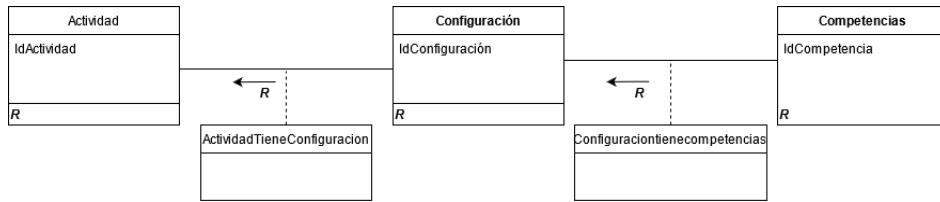


Figura 5.3.: Esquema de navegación.

- **La actividad que satisface el mayor número de tipos de actividad** En este criterio obtendríamos en orden creciente de la actividad con el mayor número de tipos de actividad que tiene la actividad. Este sería nuestro esquema de navegación:

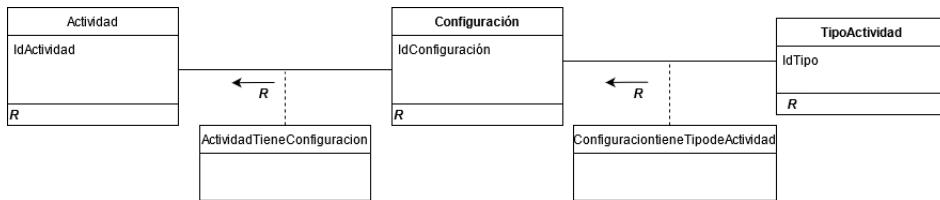


Figura 5.4.: Esquema de navegación.

## 5.4. Incorporación de nuevos ranking

Gracias a la estructura que hemos creado para construir estas sentencias, es fácil diseñar e implementar un nuevo ranking.

Lo primero que debemos hacer es pensar que queremos ordenar y si es posible con la estructura que tiene nuestra base de datos. Cuando tengamos construida esta sentencia, creamos una view con ella, para trabajar con ella con mayor facilidad. Desde la aplicación, desde la actividad de búsqueda creamos modificamos ligeramente la interfaz y el sistema para que podamos usarla.



## 6. Implementación del sistema

En este [6](#) capítulo se aborda la fase de implementación del proyecto. Se explicarán algunos detalles relevantes así como las diferentes dificultades que se encontraron en el proceso.

En esta fase se implementa el diseño realizado en el capítulo anterior de diseño, basado en el catálogo de requisitos del Capítulo [4](#) - Análisis y modelado. Como se ha especificado en el capítulo anterior, se utilizará Android studio, junto a JDBC para la conexión de la parte del servidor. Por otro lado, se utilizará el sistema operativo Windows [10](#) para trabajar con PgAdmin [4](#), con la finalidad de usar PostgreSQL.

### 6.1. Bibliotecas de Android Studio usadas durante la implementación del sistema.

Para el desarrollo de la aplicación, hemos usado librerías externas, para poder hacer más cómodo a la hora de poder implementar las funcionalidades del sistema. La mayoría de estas están publicadas en la plataforma Github. El resto de librerías venían por defecto en la aplicación.



```
dependencies {
    //librerias externas para el proyecto.
    implementation 'com.squareup.picasso:picasso:2.71828'
    implementation 'com.github.bumptech.glide:glide:4.12.0'
    implementation 'org.postgresql:postgresql:42.2.5.jre7'
    implementation 'com.github.bumptech.glide:glide:4.12.0'
    implementation "androidx.recyclerview:recyclerview:1.2.0"
    implementation 'com.google.android.material:material:1.2.1'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    //librerias internas
}
```

Figura 6.1.: Bibliotecas usadas durante la implementación.

1. Para la conexión entre nuestra aplicación y la base de datos, se hizo uso de la API de JDBC postgres. Esta fue descargada de un repositorio de Postgres [\[CJJR18\]](#).
2. Para mostrar imágenes en nuestra aplicación, se hizo uso de las librerías Picasso [\[cua16\]](#) y Glide [\[bum20\]](#)
3. Para mostrar los elementos de la lista que se ven en pantalla y los gestionará eficientemente de forma que no tenga que crear nuevos objetos para mostrar más elementos de la lista al hacer scroll se usará el RecyclerView [\[And21b\]](#)
4. Para el diseño de la interfaz como los materiales, el tipo de letra hemos usado la librería Material Components for Android [\[AS20\]](#) y ConstraintLayout [\[And21a\]](#)

## *6. Implementación del sistema*

### **6.2. Construcción de la base de datos**

La primera etapa de implementación del propio proyecto constituyó la construcción de la base de datos, para poder construir y asegurarnos de que el modelo de datos que propusimos es funcional.

Como se puede consultar la base de datos y con toda la documentación que hemos presentado, no es factible mostrar todas las sentencias SQL que compone nuestro sistema, por lo que mostraremos ejemplos de las tablas y vistas creadas en el sistema.

#### **6.2.1. Tablas**

Para este ejemplo mostraremos la sentencia que corresponde con la tabla Usuario.

```
CREATE TABLE public."Usuario"(
    "idUsuario" integer NOT NULL,
    "Nombre" character varying COLLATE pg_catalog."default",
    "Foto" character varying COLLATE pg_catalog."default",
    "Biografia" character varying COLLATE pg_catalog."default",
    "fecha" character varying COLLATE pg_catalog."default",
    "Direccion" character varying COLLATE pg_catalog."default",
    "Contrasenia" character varying COLLATE pg_catalog."default",
    CONSTRAINT "Usuario_pkey" PRIMARY KEY ("idUsuario")
)
```

Para mostrar información que contiene esta tabla usaríamos la siguiente sentencia.

```
SELECT "idUsuario", "Nombre", "Foto", "Biografia", fecha, "Direccion",
"Contrasenia" FROM public."Usuario";
```

Para insertar información en la respectiva tabla usaríamos la siguiente sentencia.

```
INSERT INTO public."Usuario"(
    "idUsuario", "Nombre", "Foto", "Biografia", fecha, "Direccion", "Contrasenia");
VALUES (?, ?, ?, ?, ?, ?, ?);
```

Para modificar información que contiene esta tabla usaríamos la siguiente sentencia.

```
UPDATE public."Usuario"
SET "idUsuario"=? , "Nombre"=? , "Foto"=? , "Biografia"=? , fecha=? ,
"Direccion"=? , "Contrasenia"=? WHERE <condition>;
```

#### **6.2.2. Views**

Para facilitar las consultas complejas, por ejemplo, la construcción de sentencias para los ranking y otras sentencias, se han creado una serie de vistas a partir de las tablas que tenemos.

```
(SELECT public."Actividad"."idActividad",
COUNT(public."usuarioformaparteactividad"."idUsuario") AS rank
```

```
FROM public."Actividad" join
public."usuarioformaparteactividad" on
public."usuarioformaparteactividad"."idActividad"=public."Actividad"."idActividad"
GROUP BY public."Actividad"."idActividad")
```

### 6.3. Interfaz

En este apartado se enumeran los aspectos más relevantes y dificultades respecto a la construcción de la interfaz de usuario.

Una de las dificultades destacables de la interfaz fue la codificación de la pantalla principal de la aplicación, en concreto del barra de navegación inferior. Hubo que utilizar elementos como Navigation Drawer, Action Bar Drawer y Navigation Adapter. Con el fin de seguir los estándares de codificación Android, se declararon los elementos que forman la barra inferior en las carpetas de recursos.

```
<item
    android:id="@+id/grupo"
    android:title="Grupos"
    app:showAsAction="never"
    android:icon="@drawable/ic_grupo"
/>

<item
    android:id="@+id/migrupo"
    android:title="Mis grupos"
    app:showAsAction="never"
    android:icon="@drawable/ic_migrupo"
/>

<item
    android:id="@+id/Actividad"
    android:title="Actividades"
    app:showAsAction="never"
    android:icon="@drawable/ic_actividad"
/>
```

Figura 6.2.: Recursos para la inferior de navegación.

Otro de los aspectos importantes respecto a la interfaz de usuario fue la conexión de este menú de navegación del que hemos hablado anteriormente con la pantalla principal que a su vez está conectada con los fragmentos.

Es decir, la menú de navegación inferior está conectado a la actividad principal.

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottomNav"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:menu="@menu/bottom_nav_menu"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent">

</com.google.android.material.bottomnavigation.BottomNavigationView>
```

Figura 6.3.: Recurso para la conexión con la actividad principal.

Y para conectar la actividad con los fragmentos que representarán al menú de navegación.

## 6. Implementación del sistema

```
<FrameLayout  
    android:id="@+id/hostfragment"  
    android:layout_width="0dp"  
    android:layout_height="0dp"  
    app:layout_constraintTop_toBottomOf="@+id/card1"  
    app:layout_constraintBottom_toTopOf="@+id/bottomNav"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent" >  
  
</FrameLayout>
```

Figura 6.4.: Recurso para la conexión con la con los fragmentos.

## 6.4. Parte lógica

En este apartado se enumeran los aspectos más relevantes respecto a la construcción de la lógica del programa.

Uno de los aspectos más importante que nos planteamos desde el principio de la implementación fue el contexto del usuario, es decir, desde que entras con tu cuenta hasta como mantienes ese contexto a lo largo de la vida del programa. Esto es importante porque la mayoría de las consultas parte del o están relacionadas con ese usuario en ese momento.

Como nuestra aplicación están compuesta principalmente por actividades y fragmentos, para navegar en ellas se hace a través de la funciones Intent y startactivity. Con Intent configuramos el contexto actual y cual queremos ir, mientras startactivity envía el objeto creado.

Gracias a esto podemos, antes de ejecutar la función startactivity, añadimos los cualquier tipo de dato que queramos enviar y en nuestro caso enviamos los datos del usuario.

```
fun enviaraperfil(){  
    val i = Intent( packageContext: this, PerfilActivity::class.java)  
    var usuario=who()  
    i.putExtra( name: "idUsuario",usuario.idusuario)  
    i.putExtra( name: "Nombre",usuario.nombre)  
    i.putExtra( name: "contraseña",usuario.contraseña)  
    i.putExtra( name: "foto",usuario.foto)  
    i.putExtra( name: "biografia",usuario.biografia)  
    i.putExtra( name: "fecha",usuario.fecha)  
    i.putExtra( name: "direccion",usuario.direccion)  
    startActivity(i)  
}
```

Figura 6.5.: Cambio de actividad y envío de datos.

Para la recepción se hace algo parecido.

```

fun who():Usuario{
    val objetoIntent:Intent=intent
    var id=objetoIntent.getIntExtra( name: "idUsuario", defaultValue: 1)
    var nombre=objetoIntent.getStringExtra( name: "Nombre")
    var contraseña=objetoIntent.getStringExtra( name: "contraseña")
    var foto=objetoIntent.getStringExtra( name: "foto")
    var biografia=objetoIntent.getStringExtra( name: "biografia")
    var fecha= objetoIntent.getStringExtra( name: "fecha")
    var dirección=objetoIntent.getStringExtra( name: "direccion")

    val usuario= Usuario(id=id, nombre= nombre.toString(),
        contraseña=contraseña.toString(), foto=foto.toString(),
        biografia=biografia.toString(), fecha=fecha.toString(),
        dirección=dirección.toString())
    return usuario
}

```

Figura 6.6.: Recepción de la actividad e información

Para el caso de la comunicación de información entre una actividad y un fragmento, Android Studio deja que se puedan hacer llamadas a las funciones que pertenecen a una actividad. De esta forma, resolvimos el problema de los datos del usuario para mostrar sus grupos y actividades pertenecientes y el envío de la sentencia SQL de la búsqueda.

```

var user=(activity as MainActivity?)?.who()
var cadena=(activity as MainActivity?)?.busqueda()

```

Figura 6.7.: Llamada a las funciones de una actividad desde un fragmento

Por otro lado, para las lista de grupos y actividades se ha implementado siguiendo las pautas y directrices que marca Android Studio en sus guías oficiales, ya que es un proceso costoso y difícil la primera vez que lo implementas [daS21]



## 7. Fase de pruebas

En primer lugar, habrá una fase de pruebas para la parte del cliente [7.1](#), y por otro lado, la parte servidor de la base de datos [7.2](#). Aunque el capítulo no sea muy importante y relevante, dado que no se ha podido hacer pruebas a fondo, se ha podido encontrar errores de novato que han sido subsanados.

Para ver un ejemplo de funcionamiento de la aplicación [pulse aquí](#)

### 7.1. Pruebas en el cliente

El objetivo de este ha sido probar todas las funcionalidades que dispone la aplicación para detectar posibles fallos de programación y dar solución a ellos.

Se ha empezado buscando posibles fallos de la interfaz, en los que los botones, campos, desplegables y demás elementos que lo componen estén conectado correctamente a las funcionalidades lógicas del programa.

Entrada	Versión 1.0	Versión 1.1
Registrarse sin indicar algún campo	Se permite el registro pero dentro de la sesión produce fallos	No permite el registro
Inicio de sesión sin indicar algún campo	No permite el inicio de sesión	No permite el inicio de sesión
Inicio de sesión con unos datos no registrados	No permite el inicio de sesión	No permite el inicio de sesión
Se intenta hacer una búsqueda sin seleccionar los campos necesarios	Muestra un mensaje de error	Muestra un mensaje de error
Se intenta crear una actividad o grupo sin llenar los campos	Se permite la creación pero dentro de la sesión produce fallos	No permite la creación de la actividad o grupo

Tabla 7.1.: Tabla de pruebas de la aplicación.

De estos errores encontrados, se ha creado una segunda versión de la aplicación donde hemos podido arreglar los errores más críticos y que dan una mala experiencia al usuario.

## 7. Fase de pruebas

Entrada	Versión 1.0	Versión 1.1
Crear una actividad sin antes crear un grupo	Se permite la creación pero dentro de la sesión produce fallos	No permite la creación de la actividad hasta que se cree el grupo
Unirte a tu propio grupo o actividad	No se permite unirse	No se permite unirse
Unirte a un grupo o actividad con el máximo de personas unidas	No se permite unirse	No se permite unirse
Unirte a un grupo o actividad y que se muestre estas al instante en la lista	No se muestra el grupo o actividad al instante	Se muestra el grupo o actividad al instante al refrescar la pantalla
Introducir una url para el perfil no válida.	Da fallo al ver el perfil	No muestra nada pero no da fallo
Crear una actividad sin tener disponible al menos un establecimiento.	Se permite crear la actividad	Se permite crear la actividad
Inicio de sesión dejando vacío los campos	Permite el inicio de sesión	No permite el inicio de sesión

Tabla 7.2.: Tabla de pruebas de la aplicación.

La mayoría de las pruebas que se intentaron fueron funcionalidades no permitidas, para comprobar que el sistema lo detecta y tomar medidas al respecto. En su segunda versión se ha podido arreglar la mayoría de estos errores.

## 7.2. Pruebas en el servidor

Como se ha descrito en apartados anteriores, la parte del servidor está formado por la API de JDBC y un servidor de base de datos Postgres. Para comprobar los errores se ha usado PgAdmin 4 y las sentencias SQL que se usan en la aplicación.

Para empezar se ha comprobado que nuestro modelo de datos que diseñamos en capítulos anteriores concuerda con la estructura de nuestra base de datos. No se encontró ningún fallo crítico que tuviera que crear de nuevo la estructura o modificarla en parte.

Por otro lado, comprobamos en su mayoría sentencias SQL funcionara y que estas, devolvieran los datos, insertaran nuevos datos o hicieran alteraciones de forma correcta. Aquí no se encontraron errores de código, pero nos dimos cuenta de los siguientes problemas:

1. Devolución de datos que luego no se usaban en la aplicación.
2. El usuario a la hora de registrarse y crear su contraseña, no se cifraba por lo que un administrador de la base de datos, podía ver la contraseña de los usuarios.
3. Cada vez que se hacía una consulta se cerraba la conexión con la base de datos, provocando un rendimiento pésimo en la aplicación.

El resultado de todo llevo a la creación de un pequeño capítulo en el diseño enfocado en la seguridad y una posterior implementación en el sistema.

## **8. Conclusión y trabajos futuros**

Este capítulo se dedica a conclusiones finales y personales del proyecto, extraídas a lo largo de todo el proceso de desarrollo del mismo.

### **8.1. Conclusiones**

A modo de conclusión, me gustaría recalcar la motivación inicial con la que se iniciaba este proyecto donde se exponía la necesidad de encontrar una solución a la necesidad de querer aprender otro idioma o simplemente relacionarte con gente que comparta esas motivaciones y que además se pueda satisfacer otras competencias que dentro del marco educativo no se tienen en cuenta. Al igual que la necesidad de comunicación entre las personas para satisfacer estas necesidades. En concreto, en este curso 2020/2021 donde la pandemia global ha puesto en jaque o anulado todas estas actividades fuera de las competencias educativas y que han caído en el olvido, dado que pocas plataformas digitales pueden satisfacer estas necesidades de las personas.

Se han cumplido la mayoría de los objetivos y requisitos mencionados en los capítulos. El principal de ellos era crear una aplicación móvil para Android capaz de ofrecer los servicios descritos. Para todo esto, ha sido necesario una gran parte de las asignaturas aprendidas durante la carrera.

Para la concepción de la idea partió de la asignatura de diseño de la interfaz de usuario, donde se nos propuso un tema y se desarrolló la idea principal, además de las vistas y experiencia de usuario. Para la análisis y modelado de esta idea se ha necesitado conocimientos de la asignatura de Algebra Lineal y Estructuras Matemáticas para el desarrollo de nuestra matriz de actividades.

También se utilizaron conceptos de las asignaturas de Fundamentos de Bases de Datos y Diseño y Desarrollo de Sistemas de Información para su diseño de la arquitectura e implementación. Para la creación de los requisitos funcionales se hizo uso de la asignatura Fundamentos de Ingeniería del Software.

En el caso de la implementación del sistema, fueron necesarios los conocimientos básicos de las asignaturas de programación, como Programación y Diseño Orientado a Objetos, Fundamentos de Programación y Metodología de la Programación.

Gracias al modelo económico de la "Economía colaborativa", hemos podido crear una plataforma digital donde diferentes elementos pueden beneficiarse, rompiendo con los antiguos esquemas del mercado tradicional donde no podría beneficiarse sin esa estructura costosa que necesitaría.

El modelo de datos que hemos diseñado e implementado en nuestro sistema hace que podamos explotarlo con relativa facilidad además de poder tener una alta escalabilidad, donde el usuario si encuentra una nueva competencia, tipos de actividades o un nuevo parámetro, puede pedirle al administrador que lo inserte en la arquitectura del sistema y que pueda ser usado. A pesar de esto, la aplicación no dispone por falta de tiempo y estudio de funcionalidades para poder crear en la interfaz un modelo desplegable dinámico, que pueda explotar y mostrar todo el potencial que tiene nuestro modelo de datos.

## *8. Conclusión y trabajos futuros*

## 8.2. Desviaciones temporales y presupuestarias

Debido a que no fue posible seguir la planificación inicial, en esta sección, se muestra los retrasos producidos en la planificación inicial y por esto, el desvío en el presupuesto acordado.

En vista de los imprevistos que fueron surgiendo, especialmente debido al tiempo para el desarrollo del modelo de datos, la implementación con la base de datos y al máster que me encontraba realizando en esos meses, la planificación de ciertas fases ha requerido de aplazamientos y extensiones, como podemos observar en el siguiente diagrama que se ajusta a la realidad:

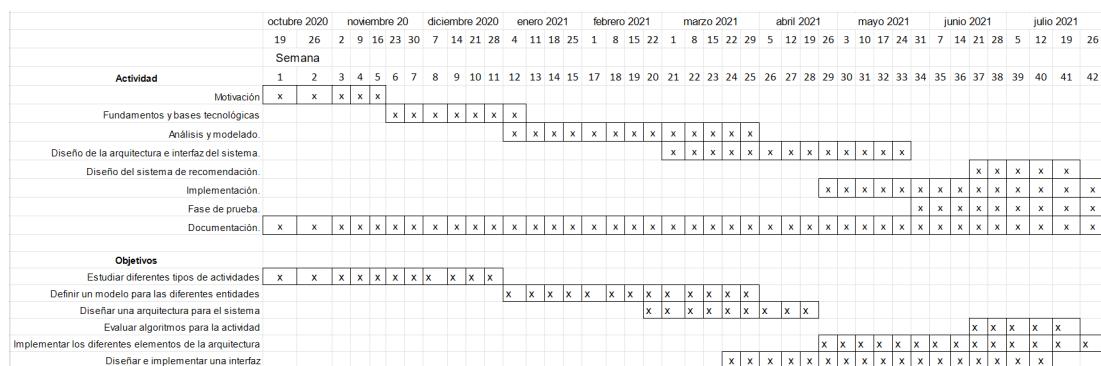


Figura 8.1.: Planificación final.

Como se puede apreciar la mayoría de los retrasos fueron producidos en las fases intermedias y finales del proyecto. Esto se debe al tiempo dedicado al modelo de datos y que el diseño del sistema de recomendación no se puso en marcha hasta que la aplicación no estaba casi en fase de pruebas. Por ello podemos ver ese salto que pega la planificación de la inicial a la final.

El proyecto se dio por cerrado a finales de julio aunque durante el mes de agosto y septiembre se llevo acabo modificaciones de la documentación y arreglos a la aplicación móvil. Además de la preparación de la presentación del proyecto.

Teniendo en cuenta todo esto, es obvio que el presupuesto que habíamos cerrado ha sufrido una desviación con el aumento de los costes. Estos coste se deben a que hemos pasado de las 39 semanas previstas a 42 semanas. Dicho esto tenemos que añadir esas 3 semanas más al proyecto. El cómputo de esta desviación presupuestaria sería de 42h extras ( $14\text{h/sem} * 3\text{ sem}$ ).

A todo esto, tenemos que añadirle que hemos usado para mantener nuestra arquitectura con Google Cloud que tendría los siguientes [costes](#). Este se incluye en herramientas de trabajo.

Concepto	Coste	Tipo de pago	Duración	Total
<b>Recursos Materiales:</b>				
<b>Asus Rog Strix</b>	899,00 €	Unico		899,00 €
<b>Sistema operativo</b>	145,00 €	Unico		145,00 €
<b>Herramientas trabajo</b>	0,00 €	Unico		0,00 €
<b>Espacio de Trabajo</b>	80,00 €	Mensual	11	880,00 €
<b>Recursos Humanos:</b>				
<b>Concepto</b>	Coste(€/h)	Tipo de pago	Horas trabajadas	Total(€)
<b>Trabajo del proyecto</b>	13,00 €	Mensual	656	8.528,00 €
<b>Tutorías del profesor</b>	40,00 €	Mensual	85	3.400,00 €
<b>Total(€)</b>				13.852,00 €

Tabla 8.1.: Costes reales.

Podemos ver, que la diferencia entre el presupuesto planeado y el real son 1008€

### 8.3. Futuros proyectos

Como futuras vías para continuar con el proyecto podríamos encontrar numerosas posibilidades que van, desde añadir más competencias, tipos de actividades, nuevos tipos de parámetros que den riqueza al proyecto, dado nuestra la arquitectura de nuestro proyecto que se pueden dividir en tres módulos puede llevarse a cabo mejoras independientemente del resto de módulos.

De cara a seguir mejorando e implementando nuevas funcionalidades que, por motivos de tiempo no han podido ser llevados a cabo, mencionaremos las siguientes posibilidades:

- Modificar la parte de la interfaz para que se puedan crear despegables dinámicos con el fin de aprovechar la escalabilidad de nuestro modelo de datos.
- Modificar la estructura de la creación de las actividades para que se puedan elegir diferentes configuraciones creadas por otros usuarios anteriormente, para esa misma actividad.
- Implementar funcionalidades con el fin de crear un monedero para poder pagar en ese momento las actividades, para poder cubrir al completo nuestra plataforma colaborativa.
- Dado que nuestra aplicación está destinado a los idiomas, la implementación de funcionalidades para poder cambiar el idioma al deseado.
- Mejoras en el sistema de recomendación, a la hora profundizar en la necesidades de los usuarios.
- Llevar a cabo un diseño más profesional de la interfaz de la aplicación, donde se muestre una verdadera experiencia para el usuario.
- Optimizaciones en la implementación con la base de datos, en concreto a través de consultas optimas.
- Dada la arquitectura que tenemos en nuestro proyecto, la posibilidad de crear una API para que se encargue de gestionar las peticiones de la base de datos.



## A. Primer apéndice

### A.1. Capa de datos del diseño de la arquitectura.

1. La tabla **Competencias** almacena la información correspondiente a las competencias que el administrador ha insertado.

Número	Columnas	Tipo	Restricciones	Descripción
1	idCompetencias	Int	PRIMARY KEY	Id de la competencia.
2	nombre	character varying		Nombre de la competencia.

Tabla A.1.: Información de la tabla Competencias.

2. La tabla **Configuración** almacena información de las configuraciones que ha creado el usuario.

Número	Columnas	Tipo	Restricciones	Descripción
1	idConfiguracion	Int	PRIMARY KEY	Id de la configuración.

Tabla A.2.: Información de la tabla Configuracion.

3. La tabla **Grupos** almacena información correspondiente a los grupos creados por los usuarios, es decir, la información necesaria para que el usuario conozca el los datos básicos del grupo.

Número	Columnas	Tipo	Restricciones	Descripción
1	idGrupo	Int	PRIMARY KEY	Id de la configuración.
2	idUsuario	Int	FOREIGN KEY	Id del usuario que ha creado el grupo.
3	nombre	character varying		Nombre del grupo
4	fecha	character varying		Fecha de creación del grupo.
5	proposito	character varying		Descripción del grupo.
6	urlgrupo	character varying		Url de la foto del grupo.

Tabla A.3.: Información de la tabla Grupos.

4. La tabla **Lugar** almacena los lugares, localizaciones que han sido creados por los usuario para que esos puedan ser usados por otros usuarios.

A. Primer apéndice

Número	Columnas	Tipo	Restricciones	Descripción
1	idLugar	Int	PRIMARY KEY	Id de la configuración.
2	idUsuario	Int	FOREIGN KEY	Id del usuario que ha creado el lugar.
3	nombre	character varying		Nombre del lugar
4	localización	character varying		Lugar exacto del lugar.
5	precio	double precision		Precio del lugar que se le ha asignado.
6	Fecha	character varying		Fecha de creación del lugar.

Tabla A.4.: Información de la tabla Lugar.

5. La tabla **Parámetro** almacena información de todos los parámetros que el administrador ha insertado.

Número	Columnas	Tipo	Restricciones	Descripción
1	idParametro	int	PRIMARY KEY	Id del parámetro.
2	nombre	character varying	NOT NULL	Nombre del parámetro
3	esSubtipoDe	int	UNIQUE/FOREIGN KEY	Id del subtipo del que está relacionado

Tabla A.5.: Información de la tabla Parámetro.

6. La tabla **TipoActividad** almacena información de todos los tipos de actividades que el administrador ha insertado.

Número	Columnas	Tipo	Restricciones	Descripción
1	idTipoActividad	Int	PRIMARY KEY	Id del tipo de actividad.
2	nombre	character varying		Nombre del tipo de actividad

Tabla A.6.: Información de la tabla TipoActividad.

7. La tabla **Usuario** almacena la información correspondiente a las cuentas de los usuarios, es decir, la información necesaria para que el usuario se identifique ante el sistema.

Número	Columnas	Tipo	Restricciones	Descripción
1	idUsuario	Int	PRIMARY KEY	Id del usuario.
2	Nombre	character varying		Nombre del usuario.
3	Foto	character varying		Url de la foto del usuario.
4	Biografía	character varying		Descripción del usuario.
5	fecha	character varying		Fecha de nacimiento.
6	Direccion	character varying		Dirección del usuario.
7	Contraseña	character varying		Constraseña para las credenciales.

Tabla A.7.: Información de la tabla Usuario.

8. La tabla **ValorDebil** almacena información de los subparámetros que se almacenan en la tabla parámetros.

A.1. Capa de datos del diseño de la arquitectura.

Número	Columnas	Tipo	Restricciones	Descripción
1	idParametro	int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla parámetro.
2	idValor	int	PRIMARY KEY	Id del valor.

Tabla A.8.: Información de la tabla ValorDebil

9. La tabla **valorEntero** se almacenan los valores de tipo entero que pertenecen a los parámetros.

Número	Columnas	Tipo	Restricciones	Descripción
1	idParametro	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla valorDebil.
2	idValor	Int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla valorDebil.
3	valor	Int	NOT NULL	Corresponde con los valores en función del parámetro.

Tabla A.9.: Información de la tabla valorEntero.

10. La tabla **valorEtiqueta** se almacenan los valores de tipo string que pertenecen a los parámetros.

Número	Columnas	Tipo	Restricciones	Descripción
1	idParametro	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla valorDebil.
2	idValor	Int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla valorDebil.
3	valor	character varying	NOT NULL	Corresponde con los valores en función del parámetro.

Tabla A.10.: Información de la tabla valorEtiqueta.

11. La tabla **valorLogico** se almacenan los valores de tipo boolean que pertenecen a los parámetros.

Número	Columnas	Tipo	Restricciones	Descripción
1	idParametro	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla valorDebil.
2	idValor	Int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla valorDebil.
3	valor	boolean	NOT NULL	Corresponde con los valores en función del parámetro.

Tabla A.11.: Información de la tabla valorLogico.

12. La tabla **valorReal** se almacenan los valores de tipo real que pertenecen a los parámetros.

A. Primer apéndice

Número	Columnas	Tipo	Restricciones	Descripción
1	idParametro	int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla valorDebil.
2	idValor	int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla valorDebil.
3	valor	Double precision	NOT NULL	Corresponde con el valor.

Tabla A.12.: Información de la tabla valorReal.

13. La tabla **ConfiguracionTieneCompetencias** almacena información procedente de las competencias y la configuración.

Número	Columnas	Tipo	Restricciones	Descripción
1	idCompetencias	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla competencias.
2	idConfiguracion	Int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla configuración.

Tabla A.13.: Información de la tabla ConfiguracionTieneCompetencias.

14. La tabla **ConfiguracionTieneTipoActividad** almacena información procedentes de los tipos de actividad y la configuración.

Número	Columnas	Tipo	Restricciones	Descripción
1	idConfiguracion	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla configuración.
2	idTipoActividad	Int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla tipoActividad.

Tabla A.14.: Información de la tabla ConfiguracionTieneTipoActividad.

15. La tabla **ConfiguracionTieneValorParametro** almacena información procedentes del valor del parámetro y la configuración.

A.1. Capa de datos del diseño de la arquitectura.

Número	Columnas	Tipo	Restricciones	Descripción
1	idConfiguracion	int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla configuración.
2	idParámetro	int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla valorDebil.
3	idvalor	Double precision	PRIMARY KEY	Id del idValor. Este hereda de la tabla valorDebil.

Tabla A.15.: ConfiguracionTieneValorParametro.

16. La tabla **GrupoTieneParametro** almacena información procedentes del grupo y los parámetros. Estos les dan información adicional para describir el grupo.

Número	Columnas	Tipo	Restricciones	Descripción
1	idGrupo	Int	PRIMARY KEY/FOREIGN KEY	Id del grupo. Este hereda de la tabla Grupo.
2	idParametro	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla Parámetro.

Tabla A.16.: Información de la tabla GrupoTieneParametro.

17. La tabla **GrupoTieneValorParametro** almacena información procedentes del grupo y del valor del parámetro. Estos les dan información adicional para describir el grupo.

Número	Columnas	Tipo	Restricciones	Descripción
1	idGrupo	Int	PRIMARY KEY/FOREIGN KEY	Id de la grupo. Este hereda de la tabla Grupo.
2	idParametro	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla ValorDebil.
3	idValor	Int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla ValorDebil.

Tabla A.17.: Información de la tabla GrupoTieneValorParametro.

18. La tabla **LugarTieneParametro** almacena información procedentes del lugar y los parámetros. Estos les dan información adicional para describir el lugar que se ha creado.

Número	Columnas	Tipo	Restricciones	Descripción
1	idLugar	Int	PRIMARY KEY/FOREIGN KEY	Id de la lugar. Este hereda de la tabla Lugar.
2	idParametro	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla Parámetro.

Tabla A.18.: Información de la tabla LugarTieneParametro.

A. Primer apéndice

19. La tabla **LugarTieneValorParámetro** almacena información procedentes del lugar y del valor del parámetro. Estos les dan información adicional para describir el lugar creado.

Número	Columnas	Tipo	Restricciones	Descripción
1	idLugar	Int	PRIMARY KEY/FOREIGN KEY	Id de la lugar. Este hereda de la tabla Lugar.
2	idParametro	Int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla ValorDebil.
3	idValor	Int	PRIMARY KEY/FOREIGN KEY	Id del valor. Este hereda de la tabla ValorDebil.

Tabla A.19.: Información de la tabla LugarTieneValorParámetro.

20. La tabla **actividadTieneConfiguración** almacena información procedentes del actividad y los parámetros. Estos les dan información adicional para caracterizar las actividad que se ha creado.

Número	Columnas	Tipo	Restricciones	Descripción
1	idActividad	int	PRIMARY KEY/FOREIGN KEY	Id de la actividad. Este hereda de la tabla Actividad
2	idConfiguracion	int	UNIQUE/FOREIGN KEY	Id del configuración. Este heredade la tabla configuración

Tabla A.20.: Información de la tabla actividadTieneConfiguración.

21. La tabla **configuraciontieneparametro** almacena información procedentes del de la configuración y los parámetros. Estos les dan información adicional para caracterizar las actividad que se ha creado.

Número	Columnas	Tipo	Restricciones	Descripción
1	idConfiguracion	int	PRIMARY KEY/FOREIGN KEY	Id del configuración. Este hereda de la tabla configuración
2	idParametro	int	PRIMARY KEY/FOREIGN KEY	Id del parámetro. Este hereda de la tabla Parámetro

Tabla A.21.: Información de la tabla configuraciontieneparametro

22. La tabla **seRealiza** almacena información procedentes del lugar y las actividades. Estos relacionan la actividad y su lugar donde se hace.

Número	Columnas	Tipo	Restricciones	Descripción
1	idLugar	Int	UNIQUE/FOREIGN KEY	Id del lugar. Este hereda de la tabla Lugar.
2	idActividad	Int	PRIMARY KEY/FOREIGN KEY	Id de la actividad. Este hereda de la tabla Actividad.

Tabla A.22.: Información de la tabla seRealiza.

23. La tabla **usuarioformaparteactividad** almacena información procedentes del usuario y las actividades. Estos nos relacionan los usuarios y la actividad a la que pertenece.

Número	Columnas	Tipo	Restricciones	Descripción
1	idActividad	Int	PRIMARY KEY/FOREIGN KEY	Id de la actividad. Este hereda de la tabla Actividad.
2	idUser	Int	PRIMARY KEY/FOREIGN KEY	Id de la usuario que se une. Este hereda de la tabla Usuario.

Tabla A.23.: Información de la tabla usuarioformaparteactividad.

24. La tabla **usuarioformapartedelgrupo** almacena información procedentes del usuario y los grupos. Estos nos relacionan los usuarios y el grupo al que pertenece.

Número	Columnas	Tipo	Restricciones	Descripción
1	idGrupo	Int	PRIMARY KEY/FOREIGN KEY	Id del grupo que pertenece. Este hereda de la tabla Grupo.
2	idUser	Int	PRIMARY KEY/FOREIGN KEY	Id de la usuario que se une. Este hereda de la tabla Usuario.

Tabla A.24.: Información de la tabla usuarioformapartedelgrupo.

## A.2. Manual de uso del jdbc postgres.

La arquitectura de nuestro sistema, en el que conectamos nuestra aplicación a una base de datos SQL, hicimos la elección de postgres SQL. Para ello se hizo uso de una API llamada JDBC. Para ello se importó una librería que podemos encontrarla en la página oficial de Postgres SQL.

Para su uso se tuvo que pasar de Java a Kotlin dado que nadie, hasta donde yo he podido encontrar, lo había utilizado aun. Para ello se creó una clase llamada **conect**. Dentro de ella se tiene una configuración, para conectarse a una base de datos Postgres SQL almacenado en la nube de Google.

```
class conect {
```

#### *A. Primer apéndice*

```
private var connection: Connection? = null  
  
private val host = "35.189.243.158"  
  
private val database = "postgres"  
private val port = 5432  
private val user = "postgres"  
private val pass = "eee"  
private var url = "jdbc:postgresql:// %s:%d/%s"  
private var status = false  
  
}
```

En el caso que se utilice la base de datos alojada en la nube deberá modificar estos datos. Muchos de los datos son a simple vista, cuales se deben cambiar. Como es el caso del host, nombre de la base de datos pero en el caso de la url, deberemos buscar en las página de JDBC la apropiada en función de la base de datos que usemos. Lo mismo pasaría en la función.

```
Class.forName("org.postgresql.Driver")
```

Para ampliar los conocimientos consulte la [página oficial](#).

*A.3. Wireframes:Bocetos de la interfaz.*

### **A.3. Wireframes:Bocetos de la interfaz.**

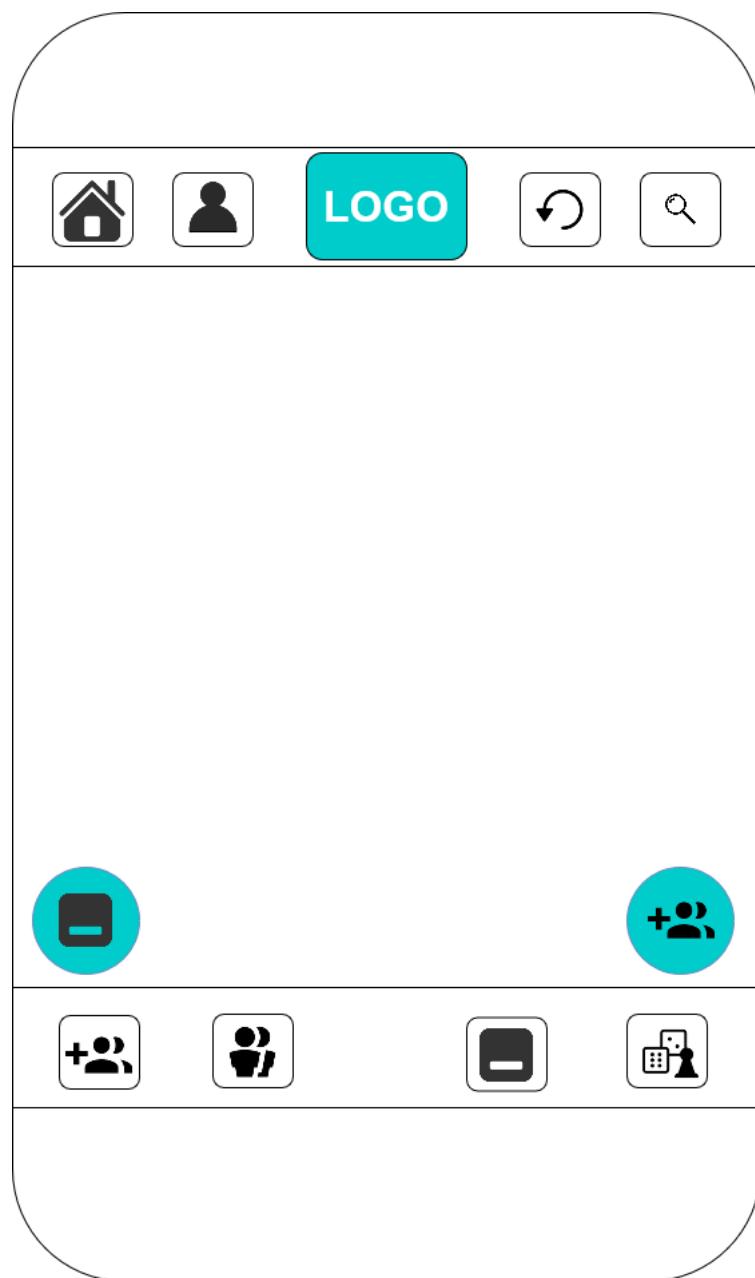


Figura A.1.: Vista principal de la aplicación.

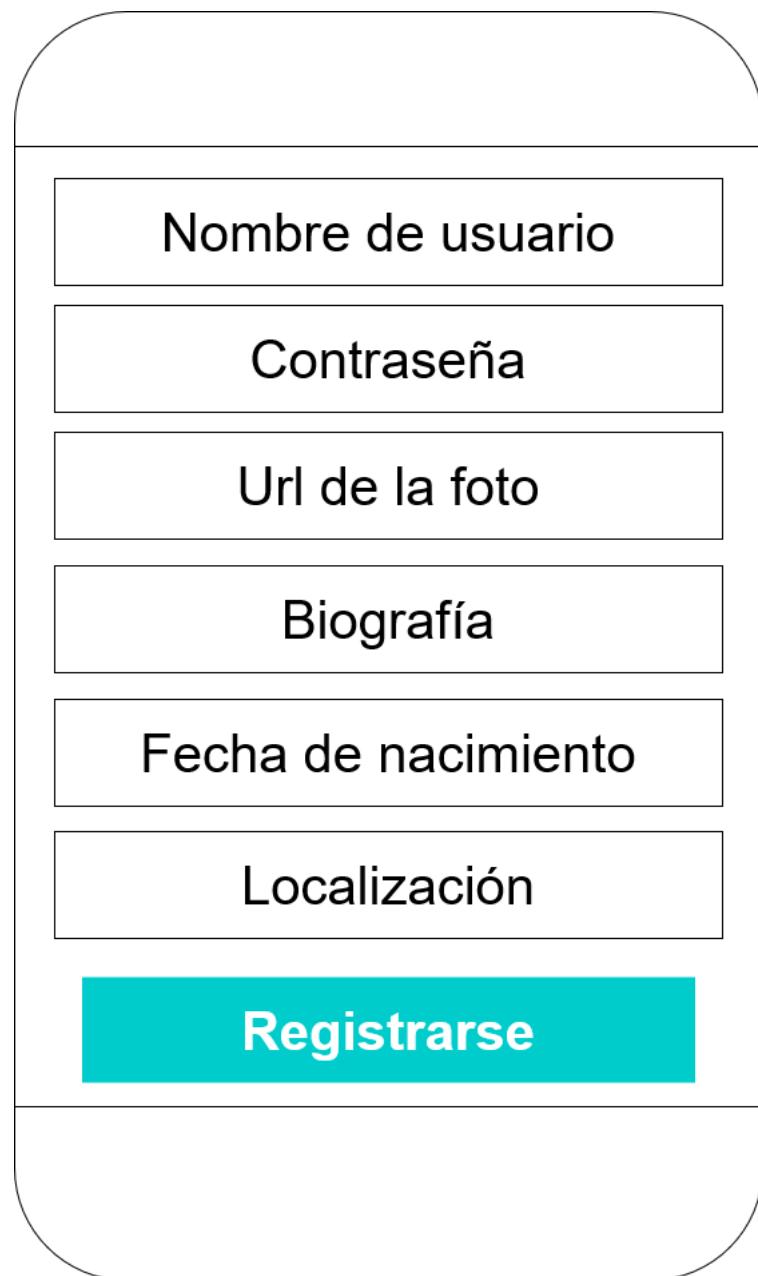


Figura A.2.: Vista del registro del usuario.



Figura A.3.: Vista del inicio de sesión.

A. Primer apéndice



Figura A.4.: Vista del perfil del usuario.



The wireframe illustrates a mobile application interface for creating a group. It features a large rounded rectangular container with several input fields and dropdown menus. At the top is a field for 'Nombre del grupo'. Below it are two dropdown menus: 'IDIOMAS' and 'Tamaño del grupo', each with a downward arrow icon. The next section contains three input fields: 'Proposito del grupo', 'Url de la foto del grupo', and 'Fecha de creación'. A prominent teal-colored button at the bottom is labeled 'Crear grupo'.

Figura A.5.: Vista de creación del grupo.

A. Primer apéndice

Nombre del establecimiento

Localización del lugar

Precio establecido

Fecha de creación

Disponibilidad

Tipo de lugar

Crear establecimiento

Figura A.6.: Vista de creación del establecimiento.

The wireframe illustrates a user interface for creating an activity. It features a large input field at the top labeled "Nombre de la actividad". Below this, there is a vertical list of eight parameters, each with a dropdown arrow icon: "Grupo", "Tamaño de la actividad", "Lugar de la actividad", "Duración de la actividad", "Relación de la actividad", "Competencias", "Tipo de actividad", and "Parámetro extra". Following these fields is a section labeled "Descripción de la actividad". Below that is another section labeled "Fecha de la actividad". At the bottom of the interface is a prominent teal-colored button labeled "Crear actividad".

Figura A.7.: Vista de creación de la actividad.

A. Primer apéndice

The diagram illustrates a search interface with the following components:

- Busqueda avanzada**: Advanced search section containing two options:
  - Combinación
  - Intersección
- Categorías**: Category section listing ten items, each with a checkbox and a dropdown arrow:
  - Idioma
  - Lugar
  - Tamaño de la actividad
  - Idioma
  - Fecha de la actividad
  - Duración de la actividad
  - Lugar de la actividad
  - Tipo de actividad
  - Competencias
- Ranking**: Ranking section listing four items, each with a checkbox:
  - Ranking 1
  - Ranking 2
  - Ranking 3
  - Ranking 4
- Buscar**: A large teal-colored search button.

Figura A.8.: Vista de la búsqueda.

A.3. Wireframes:Bocetos de la interfaz.



Figura A.9.: Vista de los grupos

A. Primer apéndice



Figura A.10.: Vista de los grupos pertenecientes.

A.3. Wireframes:Bocetos de la interfaz.

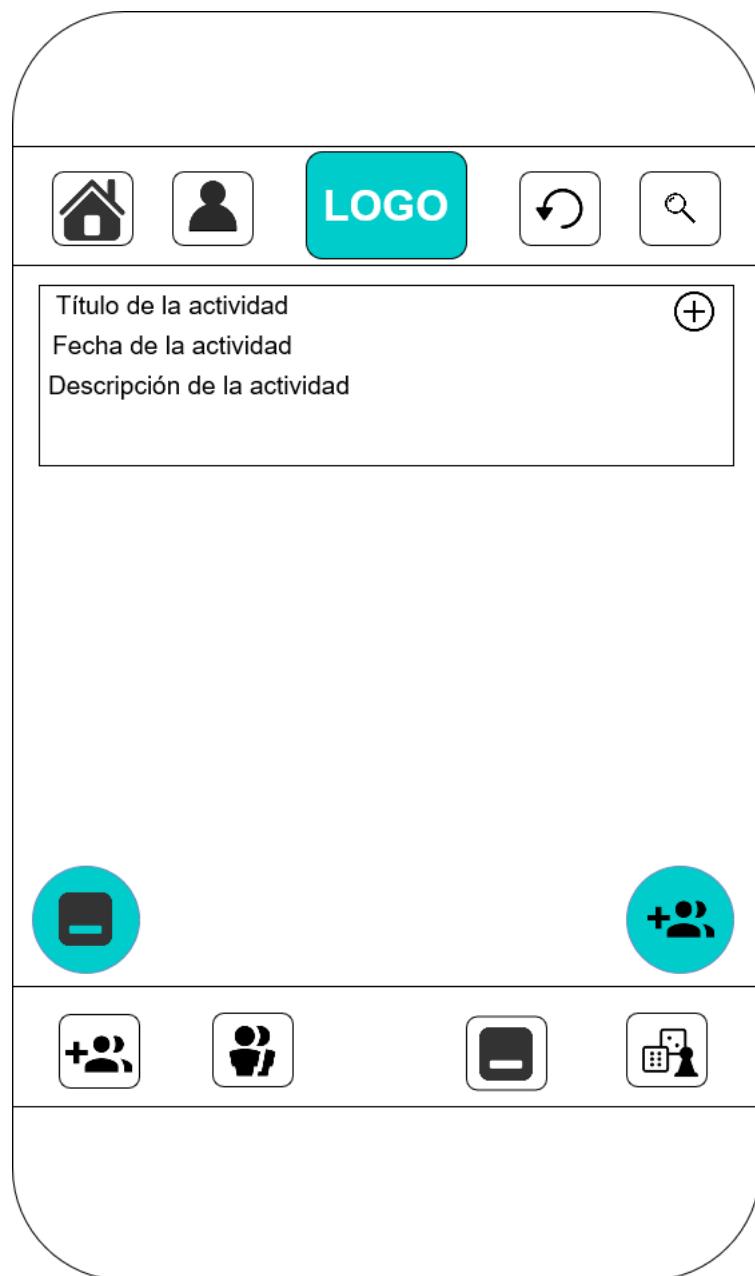


Figura A.11.: Vista de las actividades.

A. Primer apéndice

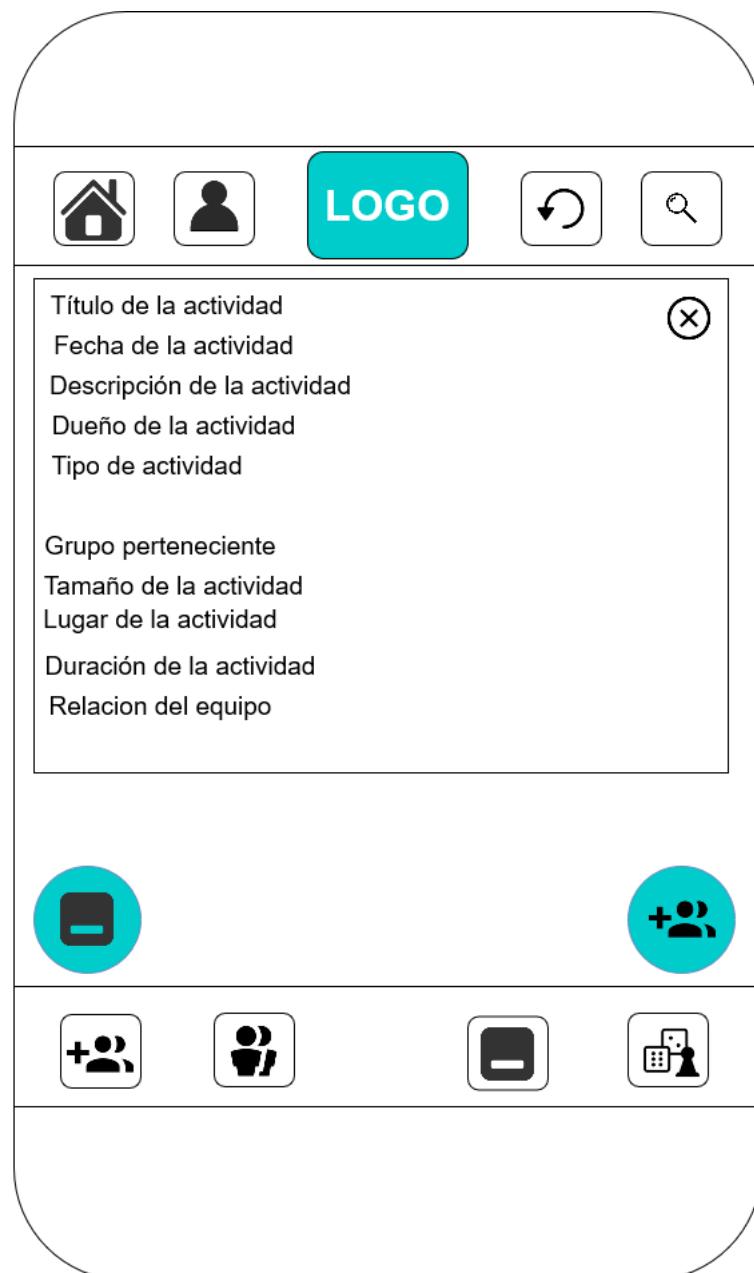


Figura A.12.: Vista de las actividades pertenecientes.

#### A.4. Manual de usuario de la aplicación.

## A.4. Manual de usuario de la aplicación.

Al arrancar la aplicación se mostrará una pantalla de inicio.

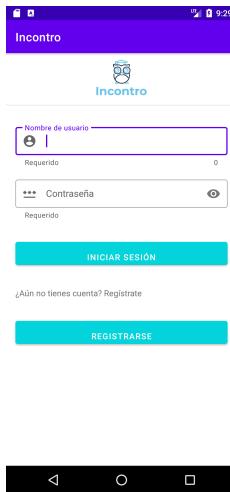


Figura A.13.: Pantalla de inicio.

A continuación, la aplicación pedirá que nos registremos, o en su defecto que iniciemos sesión.

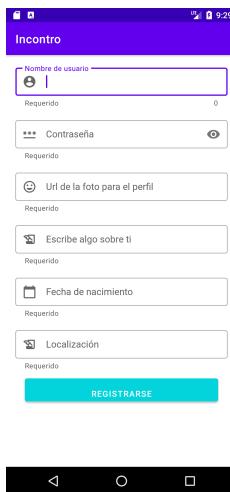


Figura A.14.: Pantalla de registro.

Durante el registro es importante que rellenemos todos los datos. Una duda que puede surgir durante el registro, puede ser el significado del tercer campo. Para que el usuario use una foto de perfil, lo que hacemos es que busque una en internet, copiando la url de esta y se copia en el tercer campo. En el campo debemos poner el formato de fecha dd/mm/aa.

#### A. Primer apéndice

Una vez iniciada la sesión, se accede a la pantalla principal, donde podemos visualizar los grupos y las actividades. Desde esta pantalla



Figura A.15.: Pantalla de los grupos.

Pulsando los botones respectivos de las listas de los grupos y actividades, podemos unirnos a ellos. No podremos unirnos si esa actividad o grupo se ha creado nosotros, ya estamos dentro al crearla, si ya nos hemos unido con anterioridad o ya se han unido el máximo número de personas.



Figura A.16.: Pantalla de las actividades.

#### A.4. Manual de usuario de la aplicación.

Desde la pantalla principal podemos acceder a la creación de un establecimiento, en ella debemos llenar los datos básicos como el nombre, que tipo de lugar, si queremos que esté disponible, el precio asignado, aunque el precio si es gratuito se debe poner 0 para evitar problemas,



Figura A.17.: Pantalla de creación de un establecimiento.

Desde la pantalla principal podemos acceder al perfil, en ella podemos ver los datos personales del usuario.



Figura A.18.: Vista del perfil.

#### A. Primer apéndice

Desde la pantalla principal podemos acceder a la creación de los grupos, en ella rellenaremos los datos suficientes e igual que en la creación del usuario usaremos la url de una foto de la web. En el campo debemos poner el formato de fecha dd/mm/aa.



Figura A.19.: Pantalla de creación de los grupos.

Desde la pantalla principal podemos acceder a la creación de las actividades. Para acceder a ella, debemos crear al menos un grupo y un establecimiento. Tras cumplir con estos requisitos, iremos seleccionando y en el caso de las competencias y el tipo de actividad iremos seleccionando cada requisito y lo añadimos una sola vez. En el campo debemos poner el formato de fecha dd/mm/aa.

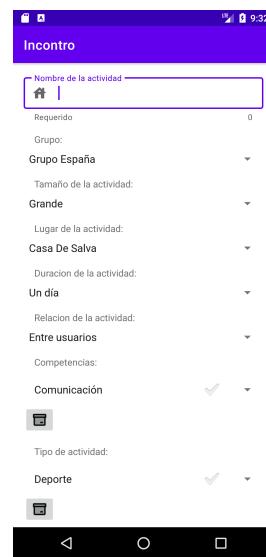


Figura A.20.: Pantalla de creación de las actividades.

#### A.4. Manual de usuario de la aplicación.

Por último, desde la pantalla principal podemos acceder a la búsqueda de actividades. Tenemos tres partes:

1. Seleccionamos en función de la unión o intersección de esas categorías que vamos a seleccionar.
2. Seleccionamos las categorías que queramos, por las que queremos buscar esa actividad.
3. Seleccionamos un solo ranking para obtener esas actividades ordenadas en función de este.

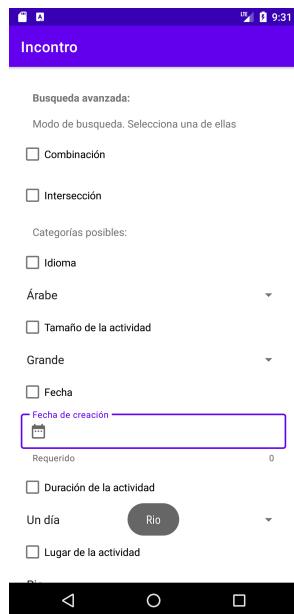


Figura A.21.: Pantalla de búsqueda de actividades.

## A.5. Manual de instalación de la aplicación.

El entorno de trabajo utilizado para la implementación de la aplicación está compuesto por todos los software y bibliotecas utilizadas en ella.

El primer software de trabajo a instalar fue Android Studio. El proceso de instalación consistió en descargar el propio software con la correspondiente versión, [Android Studio Arctic Fox | 2020.3.1](#), desde la página oficial del mismo.

A continuación, el siguiente software de trabajo a instalar fue pgAdmin 4 . Tal y como se hizo en los anteriores casos, se descargó el software, [PgAdmin 4 v5.0](#), desde la página oficial del mismo. Es una herramienta indispensable para gestionar y administrar PostgreSQL, la base de datos de código abierto más avanzada del mundo.

Por último, para poder trabajar desde diferentes ordenadores sin tener que tener instalado la base de datos, hemos alojado en <https://cloud.google.com/sql?hl=esgoogle> cloud nuestra base de datos.

Cuando tengamos los software instalados, debemos abrir PgAdmin 4 e importar el fichero SQL, para que se creen las respectivas tablas, views e inserciones con los datos para su posterior uso.

Lo siguiente que haríamos sería abrir Android Studio y con el abrir el proyecto para que podemos comenzar a usarlo. Todos los datos serán suministrados por el alumno del proyecto.

Sería suficiente solo teniendo el proyecto y Android Studio, dado que hay disponible un servidor de base de datos desplegado en la nube. En el caso que prefiera probar la base de datos con la aplicación necesitará seguir los pasos anteriores y consultar el manual de uso JDBC.

Siguiendo estos pasos, podrá iniciar la aplicación.

## Bibliografía

Las referencias se listan por orden alfabético. Aquellas referencias con más de un autor están ordenadas de acuerdo con el primer autor.

- [ABP] Aprendizaje basado en problemastécnicas didácticas. [http://www.itesca.edu.mx/documentos/desarrollo\\_academico/ Metodo\\_de\\_Aprendizaje\\_Basado\\_en\\_Problemas.pdf](http://www.itesca.edu.mx/documentos/desarrollo_academico/ Metodo_de_Aprendizaje_Basado_en_Problemas.pdf). [Citado en pág. 10]
- [act] Concepto de actividad. <https://deconceptos.com/general/actividad>. [Citado en pág. 5]
- [Ahh] 22 metodologías innovadoras que todo profesor debería conocer. <https://webdelmaestroc mf.com/portal/8-metodologias-profesor-deberia-conocer-ahora/>. [Citado en pág. 10]
- [ALJS17] Kristina Melin Anna Liv Jonsson and Ella Stensson. chapter Uppsala student nations, pages 145–164. Editorial Universidad de Granada, 2017. [Citado en pág. 18]
- [And21a] AndroidDev. Constraintlayout. <https://developer.android.com/jetpack/androidx/releases/constraintlayout>, 6 2021. Recurso online. Accedido el 9 de junio de 2021. [Citado en pág. 95]
- [And21b] AndroidDev. Recyclerview. <https://developer.android.com/jetpack/androidx/releases/recyclerview>, 6 2021. Recurso online. Accedido el 15 de junio de 2021. [Citado en pág. 95]
- [AS20] Chris Banes... Adrian Secord. Material components for android. <https://github.com/orgs/material-components/people>, 1 2020. Recurso online. Accedido el 7 de junio de 2021. [Citado en pág. 95]
- [ASG17] Sabrina Maniero Anna Serbati and Luciano Galliani. chapter The use of portfolios and reflective logs to connect learning acquired in formal and informal contexts, pages 165–182. Editorial Universidad de Granada, 2017. [Citado en pág. 19]
- [B2C20] Modelos de negocio e-commerce. <https://www.empresaactual.com/tipos-comercio-electronico/>, 2020. [Citado en pág. 24]
- [Bal11] Simone Baldassarri. Collaborative finance. <http://www.collaborativefinance.org/>, 2011. [Citado en pág. 23]
- [BCC<sup>+</sup>] I. J. Blanco, F. J. Cabrerizo, C. Cruz, M. J. Martín, M.J. Rodríguez, and D. Sánchez. Tema 2 – desarrollo de sistemas de información. [https://en.wikipedia.org/wiki/Leonhard\\_Euler](https://en.wikipedia.org/wiki/Leonhard_Euler). Recurso online. Accedido el 4 de mayo de 2019. [Citado en pág. 68]
- [Ben17] Giancarlo Benettin. chapter Galilean school of higher education, pages 125–144.. Editorial Universidad de Granada, 2017. [Citado en pág. 18]
- [BF17a] Giovanna Berizzi and Daniella Frison. chapter Vleader in the academy:Learning through an educational simulator, pages 41–57.. Editorial Universidad de Granada, 2017. [Citado en pág. 9]
- [BF17b] Giovanna Berizzi and Daniella Frison. chapter Vleader in the academy:Learning through an educational simulator, pages 47–50.. Editorial Universidad de Granada, 2017. [Citado en pág. 10]
- [BF17c] Giovanna Berizzi and Daniella Frison. chapter Learning through an educational simulator, pages 41–58,. Editorial Universidad de Granada, 2017. [Citado en pág. 16]
- [Bot20] Rachel Botsman. Rachel botsman official page. <https://rachelbotsman.com/about/>, 2020. [Citado en pág. 22]

## Bibliografía

- [bum20] bumptech. Glide. <https://github.com/bumptech/glide>, 12 2020. Recurso online. Accedido el 6 de junio de 2021. [Citado en pág. 95]
- [Che20] James Chen. Collaborative consumption. <https://www.investopedia.com/terms/c/collaborative-consumption.asp>, 2020. [Citado en pág. 20]
- [CJJR18] Dave Cramer, Kris Jurka, Oliver Jowett, and Craig Ringer. Postgres. <https://mvnrepository.com/artifact/org.postgresql/postgresql/42.2.5.jre7>, 08 2018. Recurso online. Accedido el 1 de junio de 2021. [Citado en pág. 95]
- [com] Comprador vs. consumidor, ¿a quién vendo? <https://companiademarketing.com/comprador-vs-consumidor-a-quien-vendo/>. [Citado en pág. 20]
- [com16] What is competency-based learning? <https://www.teachthought.com/learning/what-is-competency-based-learning/>, 2016. [Citado en pág. 10]
- [cua16] cuadrado. Picasso. <https://github.com/square/picasso>, 12 2016. Recurso online. Accedido el 7 de junio de 2021. [Citado en pág. 95]
- [dAa] Desarrolladores de Android. Activity. <https://developer.android.com/guide/components/activities/intro-activities?hl=es-419>. Recurso online. Accedido el 2 de mayo de 2021. [Citado en pág. 70]
- [dAb] Desarrolladores de Android. Android studio xml. <https://developer.android.com/guide/topics/ui/declaring-layout?hl=es-419#load>. Recurso online. Accedido el 2 de mayo de 2021. [Citado en pág. 69]
- [dAc] Desarrolladores de Android. Android studio.concepto. <https://developer.android.com/studio/releases?hl=es>. Recurso online. Accedido el 3 de mayo de 2021. [Citado en pág. 69]
- [dAd] Desarrolladores de Android. Carga el recurso xml. <https://developer.android.com/guide/topics/ui/declaring-layout?hl=es-419#load>. Recurso online. Accedido el 2 de mayo de 2021. [Citado en pág. 70]
- [dAe] Desarrolladores de Android. Cómo interpretar el ciclo de vida de una actividad. <https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es-419>. Recurso online. Accedido el 5 de mayo de 2021. [Citado en págs. xvi and 70]
- [dAf] Desarrolladores de Android. Kotlin.concepto. <https://developer.android.com/kotlin/overview?hl=es>. Recurso online. Accedido el 2 de mayo de 2021. [Citado en pág. 69]
- [daS21] Desarrolladores de android Studio. Cómo crear listas dinámicas con recyclerview. <https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=es-419>, 06 2021. Recurso online. Accedido el 1 de junio de 2021. [Citado en pág. 99]
- [Dec] Decsai. Especificación de requerimientos.diseño de bases de datos. [Citado en pág. 61]
- [eco20] Modelos de negocio e-commerce: B2b, b2c, c2b... ¿qué tipos de comercio electrónico existen? <https://www.empresaactual.com/tipos-comercio-electronico/>, 2020. Recurso online. Accedido el 13 de noviembre de 2020. [Citado en págs. 20 and 21]
- [emp16] Design thinking: qué es y cómo integrarlo en tus clases. <https://www.realinfluencers.es/2016/12/01/design-thinking-que-es-como-aplicarlo/>, 2016. [Citado en pág. 10]
- [Est16] Ricardo Estévez. La economía colaborativa y sus distintas formas. <https://www.ecointeligencia.com/2016/03/economia-colaborativa-formas/>, 2016. [Citado en pág. 22]
- [Fra17] María Inés Carbajal Francisco. chapter Entrepreneurs to change the world, pages 23–40,. Editorial Universidad de Granada, 2017. [Citado en pág. 15]
- [Gal] Javier Sánchez Galán. Proveedor. <https://economipedia.com/definiciones/proveedor.html>. [Citado en pág. 20]
- [Gir19] Valentina Giraldo. Plataformas digitales: ¿qué son y qué tipos existen? <https://rockcontent.com/es/blog/plataformas-digitales/>, 2019. [Citado en pág. 21]

- [Gooa] Google. buttons. <https://material.io/components/buttons>. Recurso online. Accedido el 2 de Junio de 2021. [Citado en pág. 67]
- [Goob] Google. Cards. <https://material.io/components/cards#anatomy>. Recurso online. Accedido el 2 de Junio de 2021. [Citado en pág. 67]
- [Gooc] Google. Components. <https://material.io/components?platform=android>. Recurso online. Accedido el 2 de Junio de 2021. [Citado en pág. 67]
- [Good] Google. Snackbars. <https://material.io/components/snackbars>. Recurso online. Accedido el 2 de Junio de 2021. [Citado en pág. 67]
- [Gooe] Google. Text files. <https://material.io/components/text-fields>. Recurso online. Accedido el 2 de Junio de 2021. [Citado en pág. 67]
- [Groa] The PostgreSQL Global Development Group. Postgresql 13.3 documentation. <https://www.postgresql.org/docs/13/index.html>. Recurso online. Accedido el 5 de mayo de 2021. [Citado en pág. 71]
- [Grob] The PostgreSQL Global Development Group. The postgresql jdbc interface. <https://jdbc.postgresql.org/documentation/head/index.html>. Recurso online. Accedido el 8 de mayo de 2021. [Citado en pág. 70]
- [HAB] habilidad y destreza. <https://es.scribd.com/doc/26619563/habilidad-y-destreza>. [Citado en pág. 6]
- [Hie] Janne Hietala. What is collaborative learning? <https://www.valamis.com/hub/collaborative-learning>. [Citado en pág. 10]
- [ind20] ¿cuánto se gana en españa de programador/a junior? <https://es.indeed.com/career/programador-junior/salaries>, 2020. [Citado en pág. 4]
- [JG] Pablo Beneitone Julia González, Robert Wagenaar. Un proyecto de las universidades. <https://rieoei.org/historico/documentos/rie35a08.htm>. [Citado en pág. 5]
- [JS98] Gonzalo Vazquez y Antoni Colom Jaume Sarramona. La educación no formal y otros conceptos próximos, 1998. [Citado en págs. xv, 11, and 12]
- [JWO17] Molly Kenny John Walsh, Katie Byrne and Ciara O'Farrell. chapter The class representative system of trinity college Dublin students, pages 77–88,. Editorial Universidad de Granada, 2017. [Citado en pág. 17]
- [MAK12] Maker movement. <https://www.techopedia.com/definition/28408/maker-movement>, 2012. [Citado en pág. 23]
- [Mar14] Rocío Belén Martín. Contextos de aprendizaje: formales, no formales e informales. pages 1–4, 2014. [Citado en pág. 11]
- [Mej19] Luis Miguel Arteaga Mejía. ¿qué es el software libre? <https://www.gnu.org/philosophy/free-sw.es.html>, 2019. [Citado en pág. 21]
- [met] Estrategias metodológicas. [https://cfp.us.es/web/elearning/guia/\\_10.htm](https://cfp.us.es/web/elearning/guia/_10.htm). [Citado en pág. 6]
- [Mir21] Jesús García Miranda. Construir una matriz 4x4., 2021. [Citado en pág. 27]
- [nig14] Albert Ca nigueral. CONECTA, 2014. [No citado] nigueral nigueral nigueral nigueral
- [noz19a] Pablo Jarne Mu noz. *Derecho Mercatil*, chapter 3.1, pages 95–96. Reus Editorial, 2019. [Citado en pág. 20]
- [noz19b] Pablo Jarne Mu noz. *Derecho Mercatil*, chapter 1.1, pages 9–10,17–22. Reus Editorial, 2019. [Citado en pág. 22]
- [noz19c] Pablo Jarne Mu noz. *Derecho Mercatil*, chapter 1.1, pages 21–22. Reus Editorial, 2019. [Citado en pág. 22]

## Bibliografía

- [noz19d] Pablo Jarne Mu noz. *Derecho Mercatil*, chapter 1.2, pages 22–28. Reus Editorial, 2019. [Citado en pág. 24]
- [noz19e] Pablo Jarne Mu noz. *Derecho Mercatil*, chapter 2.2, pages 61–62,75–79. Reus Editorial, 2019. [Citado en pág. 24]
- [Owy15] Jeremiah Owyang. The collaborative economy body of work for corporations. <http://web-strategist.com/blog/2015/08/27/the-collaborative-economy-defined/>, 2015. [Citado en pág. 20]
- [Owy16] Jeremiah Owyang. Jeremiah owyang official page. <https://web-strategist.com/blog/2013/08/22/table-of-contents-the-collaborative-economy/>, 2016. [Citado en pág. 22]
- [P2P20] Relaciones cliente-servidor y peer to peer. <https://www.eoi.es/blogs/eoipi/relaciones-cliente-servidor-y-peer-to-peer/>, 2020. [Citado en pág. 24]
- [pla19] Plataformas colaborativas: la nueva forma de trabajar en red | match mode. <https://matchmode.es/plataformas-colaborativas-la-nueva-forma-de-trabajar-en-red/>, 2019. [Citado en pág. 21]
- [pro] Los consumidores, clientes y usuarios. <https://www.mheducation.es/bcv/guide/capitulo/8448175840.pdf>. [Citado en pág. 20]
- [PTC19] Plataformas colaborativas: la nueva forma de trabajar en red. <https://matchmode.es/plataformas-colaborativas-la-nueva-forma-de-trabajar-en-red/>, 2019. [Citado en pág. 24]
- [rae] Rae. <https://dle.rae.es/cliente>. [Citado en pág. 20]
- [Raf] María Estela Raffino. Destreza. <https://concepto.de/destreza/>. [Citado en págs. 6 and 7]
- [Rei12] Ivana Reischel. Musemap street art app — ux case study. <https://blog.prototypi.io/musemap-street-art-app-ux-case-study-9bec6a99823b>, Aug 2012. Recurso online. Accedido el 17 de Abril de 2021. [Citado en págs. 61 and 81]
- [Rev] UNIR Revista. Qué es el desarrollo cognoscitivo y sus implicaciones en el ámbito de la educación especial. <https://www.unir.net/educacion/revista/desarrollo-cognoscitivo-cognitivo/>. [Citado en pág. 5]
- [RGM17] Isabel Romero Rosa Garay and Julia M.Gonzales. chapter An experience at the colegio mayor padre Poveda in Madrid, pages 89–104,. Editorial Universidad de Granada, 2017. [Citado en pág. 17]
- [RM17] M.luz Renuncio and Julia M.Gonzales. chapter The college council, pages 105–124,. Editorial Universidad de Granada, 2017. [Citado en pág. 17]
- [rub] Learning through arguing. <https://busytteacher.org/10319-learning-through-arguing-the-keys-to-successful.html>. [Citado en pág. 10]
- [SER99] ANTONIO GUERRERO SERÓN. El enfoque de las competencias profesionales: una solución conflictiva a la relación entre formación y empleo. pages 338–339, 1999. [Citado en pág. 7]
- [Smio06] Yajahira Smitter. Hacia una perspectiva sistémica de la educación no formal. <https://www.redalyc.org/articulo.oa?id=76102213>, 2006. [Citado en pág. 11]
- [SR17] Aurelio Villa Sánchez and Manuel Poblete Ruiz. chapter Competence:Leadership., pages 308–316,. University of Deusto., 2017. [Citado en pág. 13]
- [SSF17] María de Frutos Lobo. Sebastián Sánchez Fernandez. chapter Co-Existence in Melilla, pages 59–76,. Editorial Universidad de Granada, 2017. [Citado en pág. 16]
- [tic20] Information and communications technology (ict). <https://www.techopedia.com/definition/24152/information-and-communications-technology-ict>, 2020. [Citado en pág. 21]
- [Tunoo] Team Tuning. Tuning academy.lists of competences. <http://tuningacademy.org/lists-of-competences/>, 2000. [Citado en pág. 9]

- [viv20] Crowdfunding: ¿qué es? ¿cómo funciona? <https://www.vivus.es/blog/economia-de-hoy/crowdfunding-que-es-como-funciona/>, 2020. [Citado en pág. 21]
- [Wah17] Daniel Christian Wahl. The collaborative economy body of work for corporations. <https://medium.com/age-of-awareness/collaborative-consumption-and-peer-to-peer-collaboration-d6e40c04e2e2>, 2017. [Citado en págs. xv and 23]
- [Yar17] María Yarosh. chapter Rubricas del proyecto DARE+, pages 257–259. Editorial Universidad de Granada, 2017. [Citado en pág. 14]
- [yIMP20a] Juan Salvador Molina Martín y Ignacio Morillas Padial. Prácticas diseño interfaces de usuario 2019-20 (economía colaborativa). <https://github.com/salva12345678/DIU/tree/master/P1>, 2020. [Citado en pág. 82]
- [yIMP20b] Juan Salvador Molina Martín y Ignacio Morillas Padial. Prácticas diseño interfaces de usuario 2019-20 (economía colaborativa). <https://github.com/salva12345678/DIU/blob/master/P1/Usability-review.pdf>, 2020. [Citado en pág. 82]
- [yIlo8] Pulido Trullén y Juan Ignacio. Competencias genéricas y transversales de los titulados universitarios. [https://ice.unizar.es/sites/ice.unizar.es/files/users/leteo/publicaciones/col.\\_documentos\\_08.pdf](https://ice.unizar.es/sites/ice.unizar.es/files/users/leteo/publicaciones/col._documentos_08.pdf), 2008. [Citado en pág. 7]
- [yP99] Davenport y Prusak. Datos, información, conocimiento. [https://www.sinnexus.com/business\\_intelligence/piramide\\_negocio.aspx](https://www.sinnexus.com/business_intelligence/piramide_negocio.aspx), 1999. [Citado en pág. 21]
- [yPB17] Maria Yarosh y Pablo Beneitone. chapter Introduction, pages 11–12. Editorial Universidad de Granada, 2017. [Citado en pág. 7]