

Diseño de prototipo para el control de finanzas personales mediante análisis predictivo en una aplicación móvil

Nicolás Rodríguez Gutiérrez

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería
Especialización en Ingeniería de Software
Bogotá D.C., Colombia
2021

Diseño de prototipo para el control de finanzas personales mediante análisis predictivo en una aplicación móvil

Nicolás Rodríguez Gutiérrez

Trabajo de grado presentado como requisito parcial para optar al título de:
Especialista en Ingeniería de Software

Director:
Sandro Javier Bolaños Castro

Revisor:
John Jairo Londoño Pérez

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería
Especialización en Ingeniería de Software
Bogotá D.C., Colombia
2021

Índice general

Lista de figuras IX

Lista de tablas 1

1. INTRODUCCIÓN 2

I. CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN 3

2. DESCRIPCIÓN DE LA INVESTIGACIÓN 4

2.1. Planteamiento/Identificación del problema . . . 4
2.1.1. Planteamiento del problema . . . 4
2.1.2. Formulación del problema . . . 4
2.1.3. Sistematización del problema . . . 4
2.2. Objetivos . . . 6
2.2.1. Objetivo general . . . 6
2.2.2. Objetivos específicos . . . 6
2.3. Justificación del trabajo/investigación . . . 7
2.3.1. Justificación practica . . . 7
2.4. Hipótesis . . . 8
2.5. Marco referencial . . . 8
2.5.1. Marco teórico . . . 8
2.5.2. Marco conceptual . . . 12
2.6. Metodología de la investigación . . . 16
2.6.1. Tipo de estudio . . . 16
2.6.2. Método de investigación . . . 16
2.6.3. Fuentes y técnicas para la recolección de la información . . . 16
2.6.4. Tratamiento de la información . . . 16
2.7. Organización del trabajo de grado . . . 17
2.8. Estudio de sistemas previos . . . 18

II. DESARROLLO DE LA INVESTIGACIÓN 19

3. TÉCNICAS DE ANÁLISIS PREDICTIVO 20

3.1. Regresión lineal 20
3.1.1. Implementación 21
3.2. Regresión logística 24
3.2.1. Implementación 26
3.3. Clustering 31
3.3.1. K-means clustering 32
3.3.2. Implementación 33
3.4. Arboles de decisión 37
3.4.1. Implementación 39

4. ARQUITECTURA EMPRESARIAL 44

4.1. Organización 45
4.1.1. Introducción 45
4.1.2. Empresa 45
4.1.3. Misión 45
4.1.4. Visión 45
4.1.5. Objetivos organizacionales 45
4.1.6. Procesos 46
4.1.7. Servicios 46
4.1.8. Productos 46
4.1.9. Requerimientos organizacionales 46
4.1.10. Estructura orgánica 46
4.2. Lenguaje y el método 48
4.2.1. ADM 48
4.2.2. Archimate 49
4.3. Capa de motivación 51
4.3.1. Introducción 51
4.3.2. Punto de vista de interesados 52
4.3.3. Punto de vista de realización de objetivos 54
4.3.4. Punto de vista de contribución de objetivos 56
4.3.5. Punto de vista de principios 58
4.3.6. Punto de vista de realización de requerimientos 60
4.3.7. Punto de vista de motivación 62
4.4. Capa de estrategia 64
4.4.1. Introducción 64
4.4.2. Punto de vista de mapa de capacidad 65
4.4.3. Punto de vista de realización de resultado 67

4.4.4. Punto de vista de mapa de recurso . . . 69
4.4.5. Punto de vista de flujo de valor . . . 71
4.4.6. Punto de vista de estrategia . . . 73
4.5. Capa de negocio . . . 75
4.5.1. Introducción . . . 75
4.5.2. Punto de vista de organización . . . 76
4.5.3. Punto de vista de cooperación de actor . . . 78
4.5.4. Punto de vista de función de negocio . . . 80
4.5.5. Punto de vista de proceso de negocio . . . 82
4.5.6. Punto de vista de cooperación de proceso de negocio . . . 84
4.5.7. Punto de vista de producto . . . 86
4.6. Capa de aplicación . . . 88
4.6.1. Introducción . . . 88
4.6.2. Punto de vista de comportamiento de aplicación . . . 89
4.6.3. Punto de vista de cooperación de aplicación . . . 91
4.6.4. Punto de vista de estructura de aplicación . . . 93
4.6.5. Punto de vista de uso de aplicación . . . 95
4.7. Capa de tecnología . . . 97
4.7.1. Introducción . . . 97
4.7.2. Punto de vista de tecnología . . . 98
4.7.3. Punto de vista de uso de tecnología . . . 100
4.7.4. Punto de vista de despliegue e implementación . . . 102
4.7.5. Punto de vista de estructura de información . . . 104
4.7.6. Punto de vista de realización del servicio . . . 106
4.7.7. Punto de vista físico . . . 108
4.7.8. Punto de vista de capas . . . 110
4.8. Capa de implementación y migración . . . 112
4.8.1. Introducción . . . 112
4.8.2. Punto de vista de proyecto . . . 113
4.8.3. Punto de vista de migración . . . 115
4.8.4. Punto de vista de implementación/migración . . . 117

5. PROTOTIPO DE APLICACIÓN 119
5.1. API REST (BackEnd) . . . 119
5.2. APP Móvil (FrontEnd) . . . 123
5.2.1. Componentes ingresos y gastos . . . 126
5.2.2. Componente balance . . . 127
5.2.3. Componente estadísticas . . . 134
5.3. Base de datos . . . 136
5.4. Modelo de análisis predictivo . . . 140

III. CIERRE DE LA INVESTIGACIÓN 150

6. RESULTADOS Y DISCUSIÓN 151

7. CONCLUSIONES 152
7.1. Verificación, contraste y evaluación de los objetivos 152
7.2. Síntesis del modelo propuesto 153
7.3. Aportes originales 153
7.4. Trabajos o publicaciones derivadas 153

8. PROSPECTIVA DEL TRABAJO DE GRADO 154
8.1. Líneas de investigación futuras 154
8.2. Trabajos de investigación futuros 154

Bibliografía 157

Lista de Figuras

2-1. Grafico actitud con respecto al futuro 9
2-2. Grafico horizontes de planificación financiera 9
2-3. Grafico funciones en la toma de decisiones financieras 10
2-4. Grafico planificación del uso del dinero y cumplimiento de estos planes . . . 10
2-5. Grafico precisión de la administración del dinero 11
2-6. Grafico estrategias para resolver la insuficiencia de dinero 11
2-7. Cronograma proyecto 17

3-1. Grafico de regresión lineal 22
3-2. Coeficiente de relación en regresión lineal 23
3-3. Regresión logística - variable a predecir 27
3-4. Regresión logística - variable dinero prestado 27
3-5. Regresión logística - diagrama caja distribución datos 28
3-6. Regresión logística - cantidad prestado, costo activo y edad 29
3-7. Regresión logística - edad con mas fraudes 29
3-8. Regresión logística - validación entrenamiento modelo 30
3-9. Dataset prueba k-means clustering 33
3-10.Grafico edad vs ingresos por genero 34
3-11.Grafico ingresos vs puntaje por genero 34
3-12.Grafico numero de clusters por Inertia 35
3-13.Grafico segmentación k-means 36
3-14.Estructura de un árbol [24] 37
3-15.Arboles de decisión - análisis datos 40
3-16.Arboles de decisión - variables codificadas 41
3-17.Arboles de decisión - Grafico Gini index 42
3-18.Arboles de decisión - Grafico entropía 43

4-1. Estructura Orgánica 47
4-2. ADM 49
4-3. ADM 50
4-4. Metamodelo de Motivación 51
4-5. Interesado 52
4-6. Caso Interesado 53
4-7. Realización de Objetivos 54

4-8. Caso Realización de Objetivos 55
4-9. Contribución de Objetivos 56
4-10.Caso Contribución de Objetivos 57
4-11.Principios 58
4-12.Caso Principios 59
4-13.Realización de Requerimientos 60
4-14.Caso Realización de Requerimientos 61
4-15.Motivación 62
4-16.Caso Motivación 63
4-17.Metamodelo de Estrategia 64
4-18.Mapa de Capacidad 65
4-19.Caso Mapa de Capacidad 66
4-20.Realización de Resultado 67
4-21.Caso Realización de Resultado 68
4-22.Mapa de Recurso 69
4-23.Caso Mapa de Recurso 70
4-24.Flujo de Valor 71
4-25.Caso Flujo de Valor 72
4-26.Estrategia 73
4-27.Caso Estrategia 74
4-28.Metamodelo de Estrategia 75
4-29.Organización 76
4-30.Caso Organización 77
4-31.Cooperación de Actor 78
4-32.Caso Cooperación de Actor 79
4-33.Función de Negocio 80
4-34.Caso Función de Negocio 81
4-35.Proceso de Negocio 82
4-36.Caso Proceso de Negocio 83
4-37.Cooperación de Proceso de Negocio 84
4-38.Caso Cooperación de Proceso de Negocio 85
4-39.Producto 86
4-40.Caso Producto 87
4-41.Metamodelo de Estrategia 88
4-42.Comportamiento de Aplicación 89
4-43.Caso Comportamiento de Aplicación 90
4-44.Cooperación de Aplicación 91
4-45.Caso Cooperación de Aplicación 92
4-46.Estructura de Aplicación 93
4-47.Caso Estructura de Aplicación 94

4-48.Uso de Aplicación . . . 95
4-49.Caso Uso de Aplicación . . . 96
4-50.Metamodelo de Tecnología . . . 97
4-51.Tecnología . . . 98
4-52.Caso Tecnología . . . 99
4-53.Uso de Tecnología . . . 100
4-54.Caso Uso de Tecnología . . . 101
4-55.Despliegue e Implementación . . . 102
4-56.Caso Despliegue e Implementación . . . 103
4-57.Estructura de Información . . . 104
4-58.Caso Estructura de Información . . . 105
4-59.Realización del Servicio . . . 106
4-60.Caso Realización del Servicio . . . 107
4-61.Físico . . . 108
4-62.Caso Físico . . . 109
4-63.Capas . . . 110
4-64.Caso Capas . . . 111
4-65.Metamodelo de Estrategia . . . 112
4-66.Proyecto . . . 113
4-67.Caso Proyecto . . . 114
4-68.Migración . . . 115
4-69.Caso Migración . . . 116
4-70.Implementación/Migración . . . 117
4-71.Caso Implementación/Migración . . . 118

5-1. Arquitectura Django REST Framework . . . 120
5-2. Estructura Proyecto API . . . 120
5-3. Estructura API . . . 120
5-4. API - Modelo . . . 121
5-5. API - Serializer . . . 122
5-6. API - ViewSet . . . 122
5-7. API - URL . . . 123
5-8. CRUD Django REST API . . . 123
5-9. Arquitectura React Native . . . 124
5-10.Estructura React Native . . . 124
5-11.Pantallas React Native . . . 125
5-12.App.js - StackNavigator . . . 125
5-13.Home.js - Menú principal . . . 126
5-14.App - Menú principal . . . 127
5-15.App consumo servicio listar Ingresos . . . 128

5-16.App - Listar ingresos . . . 129
5-17.App - Listar gastos . . . 130
5-18.App - Crear gasto . . . 131
5-19.App - Editar gasto . . . 132
5-20.App - Balance . . . 133
5-21.App - Estadísticas . . . 135
5-22.Base de Datos - Estructura . . . 136
5-23.Base de datos - Lógica negocio . . . 137
5-24.Base de datos - Diagrama notacional de conjuntos . . . 137
5-25.Base de datos - Diagrama entidad-relación . . . 138
5-26.Base de datos - Diagrama relacional . . . 138
5-27.DataSet Anual . . . 140
5-28.DataSet Anual . . . 141
5-29.Grafico ingresos y gastos por mes . . . 142
5-30.Grafico gasto anual por categorías . . . 143
5-31.Grafico distribución de gastos por cada categoría . . . 144
5-32.Datos de entrenamiento . . . 144
5-33.Análisis grafico datos de entrenamiento . . . 145
5-34.Histograma variable ahorro . . . 146
5-35.Tabla cruzada entrenamiento . . . 147
5-36.Árbol de decisiones gastos personales . . . 149

Lista de Tablas

3-1. Dataset regresión lineal 20
3-2. Dataset regresión logística 24
3-3. Total compras por genero 24
3-4. Matriz de distancia 32
3-5. Valores ejemplo Gini index 39

5-1. Base de datos - Dataset para análisis 139

1. INTRODUCCIÓN

A las finanzas personales y el control de gastos la mayoría de las personas no presta la atención debida, por lo tanto, es común que el gasto en muchos casos sea incensario y sobrepase los ingresos. En este contexto se dificulta tener unas finanzas sanas, así como la capacidad de concretar proyectos personales, ahorros o inversiones. En la actualidad es habitual el uso de diferentes aplicaciones móviles para la vida diaria, cubriendo múltiples necesidades mediante servicios tecnológicos. Por ejemplo, las personas han cambiado sus hábitos al momento de gestionar sus compras, al hacerlas por comercio electrónico, las relaciones familiares y círculos de amigos se fortalecen en las redes sociales e incluso la vida laboral esta frente a grandes cambios con el denominado teletrabajo. En este sentido, el presente proyecto pretende integrar las ventajas del auge de las aplicaciones y la necesidad de mejorar las finanzas personales, a través del uso de técnicas de análisis predictivo apoyadas en machine learning que permita desarrollar un prototipo de aplicación móvil que provea a la población colombiana una herramienta para el control de su comportamiento financiero mediante gestión, análisis y recomendación de gastos personales.

En el presente proyecto se analizan las técnicas de análisis predictivo mas usadas, explicando la matemática detrás de cada una y formulando un ejercicio practico en su implementación mediante el lenguaje de programación Python, con ayuda de librerías creadas para dicho fin. También se trabaja en un capitulo de arquitectura empresarial en torno a una idea de negocio y organización que respalde el desarrollo de la aplicación, por ultimo se muestra el producto final, la aplicación móvil, detallando las tecnologías usadas, la arquitectura y el detalle técnico de su realización, así como su uso. Con la implementación de esta solución de software se presenta ante la sociedad una herramienta con la que puedan llevar un control de sus gastos personales, contando con la opción de evaluar mediante el análisis de los datos suministrados en la aplicación, el estado de las finanzas personales, proyectando y analizando los gastos presentes y futuros.

Parte I.

CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

2. DESCRIPCIÓN DE LA INVESTIGACIÓN

2.1. Planteamiento/Identificación del problema

2.1.1. Planteamiento del problema

En la actualidad son muchas las personas con dificultades en la administración de sus finanzas personales, esto deriva en una problemática que afecta el bienestar de las familias al realizar gastos en función a ingresos futuros o gastar la totalidad de estos, sin generar un ahorro y negando así la posibilidad de contar con recursos de reserva para contingencias o emergencias, muchas veces dependiendo de tarjetas de crédito o prestamos con intereses excesivos. En los gastos realizados se evidencia un alto porcentaje de gastos innecesarios, esto no permite tener metas u objetivos financieros como la adquisición de bienes inmuebles o productivos, proyectos de inversión o la participación de negocios. Por lo tanto, se evidencia la necesidad de disponer de herramientas que orienten a las personas sobre sus decisiones financieras y que le brinden asesoría sobre qué tan convenientes son sus gastos. Como respuesta a la situación descrita, se creará un prototipo de aplicación que mediante técnicas de análisis predictivo implementadas mediante algoritmos facilite la revisión, el control y la toma de decisiones de gastos personales, teniendo en cuenta los ingresos y egresos clasificados por categorías, contando con las opciones de revisión del balance actual de los gastos, revisión de gastos histórico por categorías y la recomendación o sugerencia en porcentaje para los gastos que se planean realizar con base en el comportamiento financiero histórico.

2.1.2. Formulación del problema

¿Cómo mejorar mediante una aplicación móvil que utilice análisis predictivo mediante machine learning, la problemática de control de gastos personales que se presenta hoy en la sociedad?

2.1.3. Sistematización del problema

¿Como se pueden implementar técnicas de machine learning para el análisis predictivo en cuanto a decisiones de gastos personales?

¿Qué parámetros se deben tener en cuenta para llegar a un análisis y pronóstico efectivo que ayuden a la toma de decisiones financieras personales?
¿De qué manera se puede facilitar el uso de una aplicación que controle los gastos personales a la sociedad colombiana?

2.2. Objetivos

2.2.1. Objetivo general

Diseñar un prototipo de aplicación móvil para el análisis y control de finanzas personales mediante la incorporación de análisis predictivo, haciendo uso de técnicas de machine learning que permita gestionar, analizar y controlar los gastos presentes y futuros.

2.2.2. Objetivos específicos

- Indagar acerca de las técnicas de machine learning mas usadas para el análisis predictivo, haciendo uso de las variables generadas por la aplicación móvil, proponiendo un modelo de análisis predictivo dentro de la aplicación para brindar un panorama claro del control de gastos personales.
- Crear una prototipo de aplicación haciendo uso del lenguaje de programación Python para implementar el modelo de análisis predictivo, creando un API para exponer servicios consumidos por una aplicación móvil, haciendo uso de tecnologías de desarrollo para aplicaciones móviles híbridas, brindando una aplicación de control y análisis de gastos en los dispositivos móviles Android y IOS.
- Diseñar una aplicación móvil intuitiva y fácil de usar mediante el desarrollo de una interfaz gráfica agradable, sencilla, con opciones fácilmente identificables para lograr que al usuario le sea claro su uso.

2.3. Justificación del trabajo/investigación

2.3.1. Justificación practica

El presente proyecto se basa en la necesidad identificada de que no se cuenta con herramientas que ayuden a la administración, control y análisis de las finanzas personales, por lo tanto, se propone evaluar técnicas de análisis predictivo para seleccionar y verificar la mejor, de tal manera que al hacer uso de técnicas de machine learning e identificando los factores claves, se construyan los elementos y requerimientos claves para la generación de los datos suministrados a la aplicación.
El diseño de la aplicación propuesta se plantea para dispositivos móviles debido a que son los dispositivos más usados, gracias a esto se garantiza un rápido y fácil acceso a la misma, por lo tanto, se ofrecerá una herramienta para el control financiero personal con características de adaptación, interfaz simple, segura, estable y con opciones de personalización.

2.4. Hipótesis

Creando una aplicación móvil en la cual se implementen técnicas de machine learning para un análisis predictivo de gastos, basado en los datos históricos financieros y variables como ingresos y egresos de las personas se pone a disposición de la sociedad una herramienta de fácil acceso para que por medio de esta puedan orientar sus finanzas personales.

2.5. Marco referencial

2.5.1. Marco teórico

Dentro de los estudios previamente realizados se relaciona el llevado a cabo por Wilmer Garzón en su trabajo sobre el desarrollo de una aplicación móvil multiplataforma para el seguimiento y control de las finanzas personales a partir de una clasificación de ingresos y egresos, incluyendo los conceptos de ahorro y deudas; la aplicación hace un proceso de gestión de todos los movimientos que el usuario indica diariamente y se le proporciona consejos financieros suministrados a la base de datos para adquirir educación financiera; concluyen con que este proyecto brindará sus servicios para que los usuarios puedan controlar y gestionar sus finanzas personales, obteniendo una educación financiera básica. [19].
Igualmente se referencia el estudio de prefactibilidad para el diseño de una aplicación móvil para organizar y controlar las finanzas personales en Colombia, en el cual se debe diseñar una aplicación móvil que permita llevar el control de forma fácil e inmediata de las finanzas personales y que se tenga acceso a ella en cualquier momento, finalmente concluyen que existe un mercado potencial importante para el proyecto con una proyección de crecimiento continuo, identificando los recursos necesarios para desarrollar la aplicación. [1].
En cuanto al comportamiento financiero de los colombianos, el Banco Mundial y el Gobierno de Colombia realizó una encuesta de capacidades financieras y sus resultados en cuanto a la previsión para gastos futuros [32]. Se encontró que el 55 % de los encuestados realizan una planificación para el tiempo presente y no a futuro y el 39 % se enfoca a corto plazo, como lo muestra la Figura **2-1**.
Para el aspecto de planificación financiera en la encuesta se pudo concluir que el 25 % de los colombianos informa de un horizonte temporal de planificación de menos de una semana, el 21 % tiene en cuenta desde una semana hasta un mes, el 22 % considera entre uno y seis meses, y otro 22 % planifica para más de seis meses. Quienes están empleados en el sector formal utilizan horizontes temporales más prolongados, lo mismo que las personas de grupos de ingreso más alto. Un ingreso estable puede facilitar el acceso a productos y servicios; esto, a su vez, simplifica la planificación, como en el caso de las deducciones automáticas para cuentas de ahorro o de pensión, ver Figura **2-2**.
Respecto a las decisiones sobre gastos en el hogar tanto hombres como mujeres participan activamente, aunque menos mujeres reportan que aportan económicamente en el hogar, el

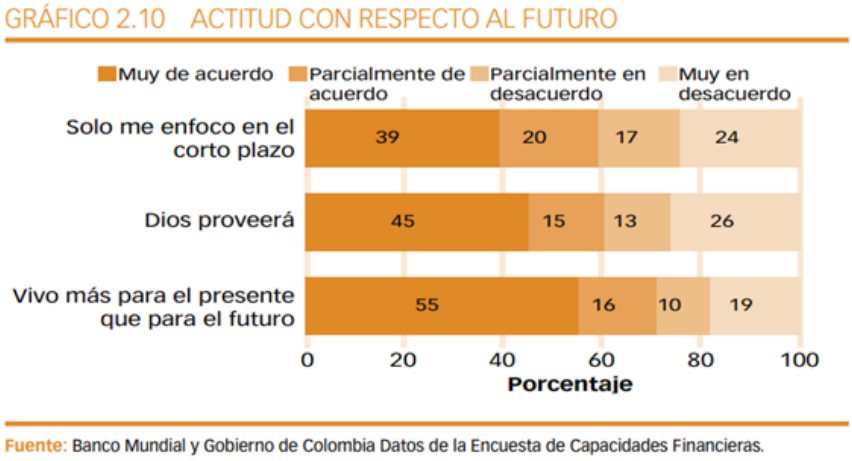


Figura 2-1.: Grafico actitud con respecto al futuro

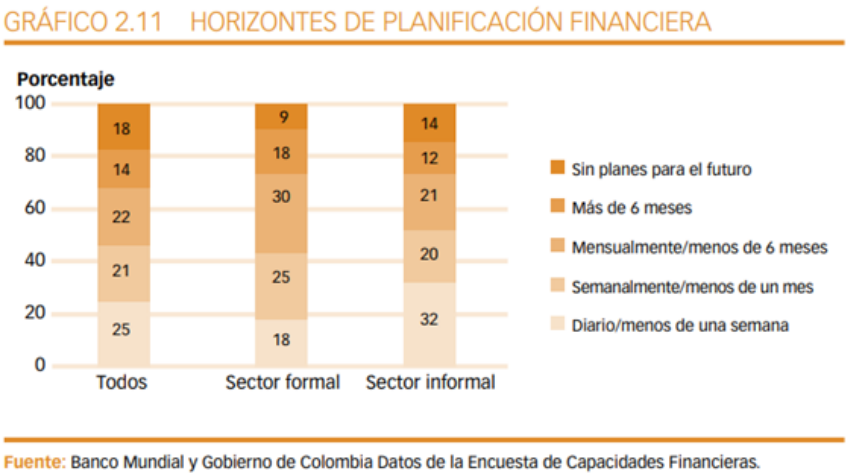


Figura 2-2.: Grafico horizontes de planificación financiera

91 % de los encuestados manifestó ser responsable de algún aspecto de los gastos del hogar, como planificar cuánto dinero se gasta, pagar las cuentas o tomar decisiones financieras por el grupo familiar, mientras que aproximadamente un 6 % es responsable solo de sus gastos personales. Y como lo muestra la siguiente gráfica, el 92 % de las mujeres dijo que participaba en la toma de decisiones financieras, ver Figura **2-3**.

Referente a la determinación del presupuesto y seguimiento de gastos, el 94 % de los adultos dijo que planificaba cómo usar su ingreso y dos tercios informaron que se ajustaban a su plan. Los adultos que planificaban se separaron en dos grupos: los que siempre planificaban y los que solo lo hacían a veces. De los encuestados, el 45 % dijo que sus planes eran exactos, el 50 % manifestó que hacía planes aproximados y el 6 % informó que no hacía ningún plan. Más de la mitad de los que no hacían planes eran adultos jóvenes de menos de 25 años

CUADRO 2.1 FUNCIONES EN LA TOMA DE DECISIONES FINANCIERAS

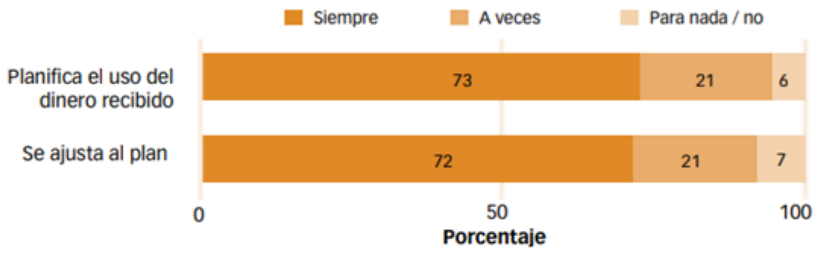
CARACTERÍSTICA	% TOMA DECISIONES FINANCIERAS DEL HOGAR	% TOMA DECISIONES FINANCIERAS PERSONALES	% NO TOMA NINGUNA DECISIÓN
Hombres	84,3	7,5	8,2
Mujeres	92,2	4,8	3,0
Edad			
18–24	76,5	23,1	0,4
25–46	92,6	4,3	3,1
47–59	91,3	1,7	7,0
≥ 60	89,0	1,5	9,5
Nivel de educación			
Primario o inferior	91,3	1,9	6,9
Secundario	89,5	6,8	3,6
Terciario	85,0	10,6	4,4

Fuente: Banco Mundial y Gobierno de Colombia Datos de la Encuesta de Capacidades Financieras.
Nota: n = 1604.

Figura 2-3.: Grafico funciones en la toma de decisiones financieras

y un 73 % se ubicaba en los dos niveles inferiores de ingreso. Mientas que el 87 % de los entrevistados informó que se ajustaba a su plan de gastos, el 67 % dijo que seguía su plan siempre, el 19 % dijo que mantenía el plan solo a veces y el 7 % nunca lo seguía, así como se aprecia en la Figura **2-4**.

GRÁFICO 2.2 PLANIFICACIÓN DEL USO DEL DINERO Y CUMPLIMIENTO DE ESTOS PLANES



Fuente: Banco Mundial y Gobierno de Colombia Datos de la Encuesta de Capacidades Financieras.

Figura 2-4.: Grafico planificación del uso del dinero y cumplimiento de estos planes

Algo fundamental y preocupante es que más de la mitad de los encuestados no sabe cuánto dispone para gastar y cuánto gastó la semana anterior, lo que indica una baja precisión en la administración diaria del dinero; de hecho, el 46 % de los entrevistados dijo que sabía de cuánto disponía su grupo familiar para gastos corrientes, solo el 24 % dijo que conocía el

monto exacto y no en forma aproximada. Y mientras que el 48 % sabía cuánto había gastado la semana anterior, solo el 23 % sabía exactamente cuánto había gastado, como lo demuestra la Figura **2-5**.

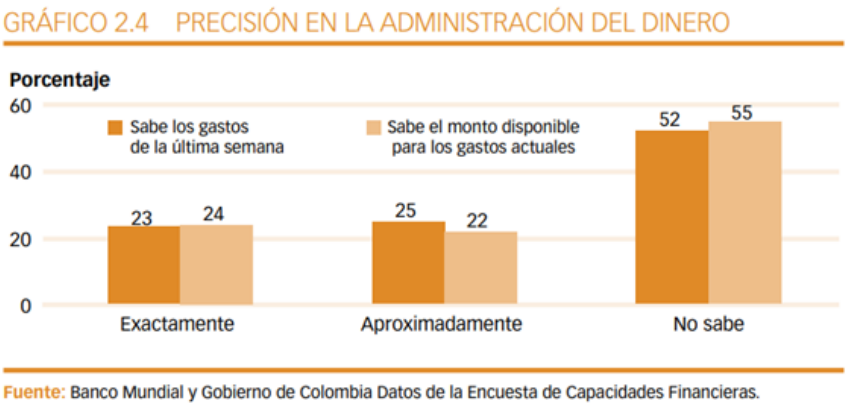


Figura 2-5.: Grafico precisión de la administración del dinero

Referente a las estrategias para resolver insuficiencia de dinero, el 28 % de los encuestados dijo que habitual u ocasionalmente pedía prestado para comprar alimentos u otros artículos de primera necesidad, mientras que el 56 % informó que pedía dinero para cubrir otras deudas. Una educación inferior a la de nivel secundario y un ingreso bajo se correlacionan con una mayor probabilidad de tener que pedir prestado para hacer frente a deudas. En general, el 53 % de quienes solicitan dinero pensaba que se había endeudado hasta llegar a su límite, mientras que el 33 % tenía la sensación de que podía pedir más. Un 14 % sentía que había pedido más de lo que podía devolver, así como lo muestra la Figura **2-6**.

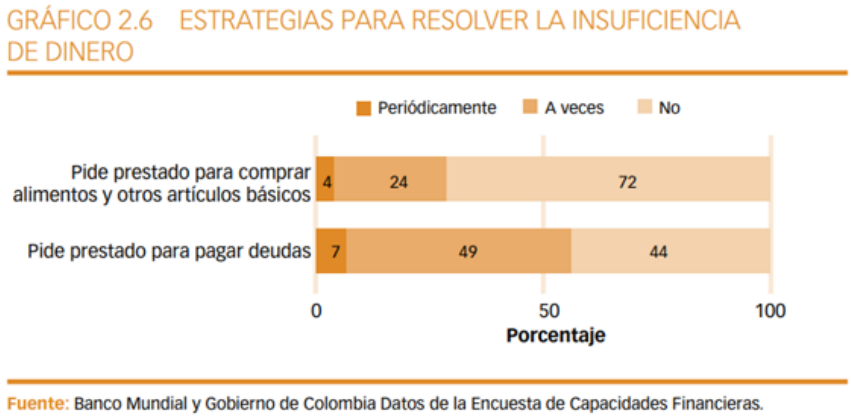


Figura 2-6.: Grafico estrategias para resolver la insuficiencia de dinero

2.5.2. Marco conceptual

Machine Learning

Las técnicas de aprendizaje automático o Machine Learning se puede definir como un conjunto de métodos capaces de detectar automáticamente patrones de datos; incluye el uso de patrones detectados para realizar predicciones o para tomar otro tipo de decisiones en entornos de incertidumbre. Igualmente es un pilar para la transformación digital gracias a la aplicación de técnicas y algoritmos capaces de aprender a partir de distintas y nuevas fuentes de información, construyendo algoritmos que mejoren de forma autónoma con la experiencia, permitiendo disponer de métodos capaces de detectar automáticamente patrones en los datos y usarlos para predecir sobre datos futuros en un entorno de incertidumbre. En cuanto a las principales tendencias para la instauración de las técnicas Machine Learning sobresale el uso de fuentes de información que recopilan datos en tiempo real y su importancia radica en obtener un poder de predicción en relación con la interpretación de los modelos, la incorporación de la capacidad de que el algoritmo se modifique de forma autónoma en función de los cambios que se van produciendo. [8].

Arquitectura de software

En la década de los 90 investigadores de la universidad Carnegie-Mellon en Pensilvania y del Instituto de Ingeniería de Software comienzan a contemplar la necesidad de investigar y desarrollar un nivel de abstracción superior al del diseño, el cual fue nombrado Arquitectura de Software; que se define como la estructura o estructuras del sistema que comprende elementos de software, las propiedades visibles externamente de estos elementos y la relación entre ellos. Siguiendo el modelo de desarrollo conocido como Modelo de Cascada, la arquitectura y el diseño se entienden como la fase de Arquitectura del Sistema la cual sigue a la Ingeniería de Requerimientos; es decir que el tener por completo los requerimientos constituye una gran ventaja para poder definir una buena arquitectura; no obstante, en la práctica esto no sucede y el diseño empieza con un conjunto incompleto de requerimientos; ya que estos van apareciendo a lo largo de la vida del sistema. En cuanto a las fases de Arquitectura del Sistema se distinguen tres etapas: elección del estilo arquitectónico, selección de los patrones de diseño y diseño de componentes. [9].

La arquitectura de software a grandes rasgos es la forma que le dan quienes construyen el sistema, esta forma es la división del sistema en componentes, como se disponen estos componentes y como se comunican con otros, el propósito de esto es facilitar el desarrollo, el despliegue, la operación y el mantenimiento del sistema de software. El objetivo principal de una arquitectura de software es la de hacer que el sistema funcione correctamente [28].

Arquitectura orientada a servicios (SOA)

SOA es Arquitectura orientada a servicios, es una visión de los sistemas de una organización como los servicios que proporcionan y no como los sistemas y funcionalidades en sí. Esta arquitectura está definida para dar soporte a servicios, por lo que, aunque no es lo mismo que los servicios web, están íntimamente relacionados, además, el auge de dichos servicios web ha supuesto también el de este tipo de arquitectura, la cual proporciona independencia total de la plataforma y del lenguaje, además de que permite generar servicios reutilizables y compartidos mediante el uso de lenguajes y protocolos totalmente estandarizados como HTTP, SOAP, XML, UDDI o WSDL. [15] La Arquitectura de desarrollo de software Orientada a Servicios viene de una evolución de orientación a objetos, luego a componentes para convertirse en un nuevo paradigma que propone convertir los recursos de software en servicios disponibles que puedan ser reutilizables y fáciles de integrar en grandes y complejas aplicaciones, facilitando la interoperabilidad entre las aplicaciones y el servicio web. Así se integran diferentes ciclos de vida e implementaciones de soluciones con múltiples tecnologías. [7] En cuanto a las principales ventajas que provee la arquitectura orientada a servicios sobresalen: brindar agilidad y flexibilidad a las organizaciones, facilitar el trabajo tercerizado, permitir la simplificación del desarrollo de soluciones mediante la utilización de estándares de la industria y capacidades comunes de industrialización, permitir el abordaje de modelos de negocios basados en colaboración con otros entes (socios y proveedores), aislar mejor los sistemas frente a los cambios generados por otras partes de la organización, promover la integración de tecnologías diferentes y además, permitir el alineamiento y acercamiento a las áreas de tecnología y negocio. [20]

Análisis descriptivo y predictivo

La minería de datos es una técnica utilizada en el Big Data para ayudar a comprender el contenido de una base datos, en un proceso analítico de exploración para descubrir patrones consistentes implícitos o relaciones sistemáticas, en la minería de datos se reúnen las ventajas de varias disciplinas como la estadística, la inteligencia artificial, la computación, las bases de datos y el procesamiento masivo, especialmente usando como materia prima las bases de datos. Tiene tres objetivos principales: descripción, predicción y prescripción. El análisis descriptivo se centra en encontrar patrones y tendencia de los datos haciéndolo interpretables por los humanos; en cuanto al análisis predictivo, se enfoca en lo que podría suceder una vez se interpreten los datos, mediante cálculos de probables eventos futuros, aplicando modelos predictivos y de probabilidad, pronósticos y análisis estadístico. En cuanto al análisis prescriptivo, se centra en proporcionar la mejor solución para el problema dado. [29]

Python Django

Django nació en el año 2003 en una agencia de prensa en Lawrence, Kansas, es un marco de trabajo web que emplea Python como lenguaje de programación para la creación de sitios web. En 2005 la agencia decidió publicar el código fuente de Django en la licencia BSD (Berkeley Software Distribution). En 2008 fue creada la Fundación de Software Django para apoyar y promover su avance; asimismo la versión 1.00 fue lanzada unos meses más tarde. El lema de Django es explícito, en el sentido que su marco de trabajo fue creado para acelerar la fase de desarrollo de un sitio, sin ser exclusivo; de hecho, este marco utiliza el patrón MVC (Modelo – Vista – Controlador) que permite tener una arquitectura coherente. En cuanto a la compatibilidad de Django, hasta el año 2013 era compatible con la versión 2X de Python; sin embargo, el 26 de febrero del mismo año se lanzó la versión 1.5 de Django y se apunta al comienzo de la compatibilidad con Python 3. [11]

Jupyter Notebook

Jupyter Notebook es una interfaz web de código abierto que permite la inclusión de texto, video, audio, imágenes, así como la ejecución de código a través del navegador en múltiples lenguajes. Esta ejecución se realiza mediante la comunicación con un núcleo (Kernel) de cálculo. Aunque en principio, el equipo de desarrollo de Jupyter Notebook incluye por defecto únicamente el núcleo de cálculo Python, el carácter abierto del proyecto ha permitido aumentar el número de núcleos disponibles, incluyendo, por ejemplo Octave, Julia, R, Haskell, Ruby, C/C++, Fortran, Java, SageMath, Scala, o también Matlab y Mathematica. Esta interfaz, agnóstica del lenguaje (de ahí su nombre al unir 3 de los lenguajes de programación de código abierto más utilizados en el ´ámbito científico: Ju-lia, Py-thon y R), puede suponer por tanto una estandarización para mostrar el contenido de cursos científicos, sin encontrarse limitado a la adopción de un único lenguaje. [13]
A diferencia de ejecutar archivos de dirección a través de Python, los Jupyter Notebook proporcionan un entorno de trabajo en el cual se emplea una interfaz basada en un navegador para una sesión interactiva de Python y así ejecutar comandos y ver los resultados y visualizaciones, los cuales también se pueden guardar en el computador. Los usuarios se benefician del resaltado de sintaxis y la finalización de tabulaciones, así como de la integración con herramientas de ciencia de datos como ROOT, Matplotlib o Pandas. [6]

Aplicaciones móviles

Una aplicación móvil es una herramienta diseñada para desarrollar una función específica en una plataforma concreta. Se pueden distinguir 3 tipos de aplicaciones móviles: la primera es la aplicación nativa que se desarrolla de forma específica para un determinado sistema operativo por lo general para desarrollo móvil con posibilidad de acceder a las características del hardware y sin la necesidad de conexión a internet; la segunda es la aplicación web que

se desarrolla con lenguajes de programación como el HTML, JavaScript y CSS y su ventaja radica en la posibilidad de programar independiente del sistema operativo en que se usará la aplicación y su ejecución se da por medio de un navegador web a través de una URL; por último, está la aplicación nativa web o aplicaciones híbridas que se desarrollan con lenguajes propios de las aplicaciones web pero con la posibilidad de acceder a características hardware del dispositivo. [26]

React Native

En la conferencia React.js en 2015, Facebook presentó su nuevo marco React Native, pensando en que revolucionaría la forma en que se crean las aplicaciones móviles. Cuando React Native fue lanzada, solo había soporte para iOS, pero desde entonces el soporte para Android se ha desarrollado y aún se está expandiendo. React Native maneja un código relativamente abierto y su principal objetivo es reducir el tiempo para crear una aplicación móvil. Si bien es necesario desarrollar dos aplicaciones para que sean compatibles con iOS y Android por las diferencias entre las apariencias, sensaciones y capacidades, no puede haber una aplicación homogénea para los sistemas operativos. Sin embargo, dado que es la interfaz gráfica la que difiere, el desarrollo podría basarse en el mismo lenguaje, pero tener los gráficos renderizados de manera diferente dependiendo de la plataforma destino y ser componentes del React Native. Facebook llama a este enfoque .ªprender una vez, escribir en cualquier lugar", que describe de qué se trata React Native; dentro de las ventajas es su ejecución en una instancia incrustada de JavaScriptCore (iOS) o V8 (Android) dentro de las aplicaciones y renderizar en componentes específicos de la plataforma de nivel superior. [10]

SQLite

En los términos más simples, SQLite es un paquete de software de dominio público que proporciona un sistema de administración de bases de datos relacionales, o RDBMS. Los sistemas de bases de datos relacionales se utilizan para almacenar registros definidos por el usuario en tablas grandes. Además del almacenamiento y la gestión de datos, un motor de base de datos puede procesar comandos de consulta complejos que combinan datos de varias tablas para generar informes y resúmenes de datos. Otros productos RDBMS populares incluyen Oracle Database, IBM, DB2 y SQL de Microsoft. Servidores comerciales con MySQL y PostgreSQL, siendo productos populares de código abierto. [23]

2.6. Metodología de la investigación

2.6.1. Tipo de estudio

El tipo de estudio del presente proyecto de investigación es de tipo descriptivo, se pretende identificar elementos y características de las técnicas de análisis predictivo, así como identificar variables financieras a tener en cuenta para poder incluirlas en el análisis predictivo y poder orientar la conducta financiera de la sociedad en general.

2.6.2. Método de investigación

Para la realización de la investigación se seguirá el método deductivo, se van a adaptar las técnicas de análisis predictivo mediante machine learning a una realidad concreta, la gestión de finanzas personales, esto gracias a la investigación de dichas técnicas y a la identificación de variables financieras a tener en cuenta para su análisis.

2.6.3. Fuentes y técnicas para la recolección de la información

En el desarrollo de la investigación se utilizarán fuentes secundarias para la información relacionada a las diferentes técnicas de análisis predictivo y machine learning, se investigará especialmente en artículos de la IEEE. Para la información relacionada a la parte financiera se utilizarán fuentes secundarias, buscando artículos del tema y otros textos para examinar las principales dificultades en cuanto a gestión financiera personal,
Para el diseño del prototipo de aplicación móvil se utilizarán fuentes primarias, mediante el uso de encuestas para definir la mejor manera de captura y presentación de información y mejorar la usabilidad según los datos estadísticos recogidos por estas encuestas.

2.6.4. Tratamiento de la información

Las fuentes de investigación secundarias utilizadas para la investigación son de carácter público, los resultados obtenidos a través de la realización del proyecto serán a su vez de divulgación pública. En cuanto a las fuentes primaras, los datos de las encuestas se capturarán de forma anónima, convirtiéndose en datos estadísticos para la realización del proyecto investigativo.

2.7. Organización del trabajo de grado

A continuación, se muestra el esquema de trabajo, mediante un cronograma usando un diagrama de Gantt, para visualizar el tiempo estimado para las diferentes etapas del proyecto **2-7**.

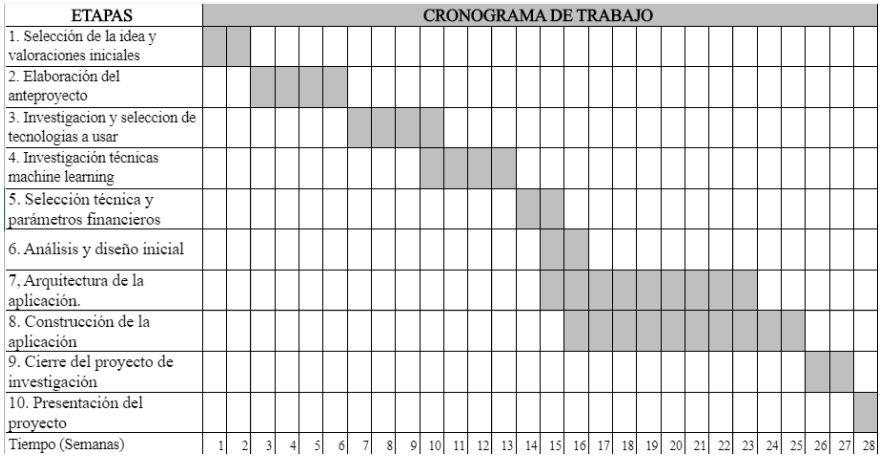


Figura 2-7.: Cronograma proyecto

2.8. Estudio de sistemas previos

Dentro del estudio adelantado de sistemas previos se encuentra los trabajos realizados acerca de diseño y desarrollo de aplicaciones para el control de finanzas personales, un ejemplo es EfecfApp, diseñado por Wilmer Garzon en el trabajo [19], en el cual ademas se muestran diferentes aplicaciones creadas para llevar un control de gastos personales, como una de control de gastos diarios, otra que sirve para generar alertas de deposito en cuenta para un sistema de ahorro, una referente a restar gastos de los ingresos, así como la propia aplicación propuesta la cual combina alguno de estos elementos identificados en las aplicaciones analizadas. Otras aplicaciones que cumplen con la finalidad de gestión de gastos personales se encuentra disponibles en las tiendas Android y IOS, como la de Toshl Finanzas [22], la cual cuenta con una funcionalidad gratuita básica para gestión de ingresos y gastos, y con opciones de pago para crear conexiones con bancos o crear presupuestos.

Al analizar aplicaciones orientadas a la gestión de finanzas personales, se encuentra que se enfocan en la administración de ingresos y gastos, así como otras interacciones como la vinculación de banco o presupuestos, algunas cuentan con gráficos de los movimientos financieros realizados, pero se ve la necesidad de un análisis mas detallado, como el propuesto en el proyecto realizado mediante un modelo de análisis predictivo haciendo uso de machine learning y datos históricos.

Como justificación a la necesidad de estas aplicaciones, se encuentra trabajos de investigación acerca de la problemática de la falta de control y gestión de estas finanzas personales por parte de la sociedad colombiana, esto basado en encuestas realizadas para dicho fin, como la encuesta nacional sobre comportamientos, actitudes y conocimientos financieros desarrollada por el banco mundial [32], detallada en el marco teórico del presente proyecto.

Parte II.

DESARROLLO DE LA INVESTIGACIÓN

3. TÉCNICAS DE ANÁLISIS PREDICTIVO

Las técnicas de análisis predictivo son modelos que se basan en conceptos estadísticos, un modelo en el contexto de la ciencia de datos intenta relacionar un conjunto de variables para dar respuestas con los datos con los que se cuenta. De manera limitada e imprecisa pero a su ves de una manera útil, un modelo nos ayuda a darle sentido a nuestro problema. [30]
En el análisis predictivo como en la estadística, se busca un significado en la relación entre variables y esas relaciones se representan en modelos, como resultado de estos modelos obtenemos otras variables de respuesta con lo que intentamos predecir, las cuales tenemos que observar, manipular o controlar para llegar a una respuesta a nuestro problema o situación inicial. [30]
A continuación, se detallaran algunas de las técnicas mas usadas para la creación de modelos de análisis predictivo, mostrando la matemática detrás de cada técnicas y su implementación en el lenguaje de programación Python, con un ejemplo ilustrativo.

3.1. Regresión lineal

La regresión lineal es una de las técnicas mas básicas y usadas para crear modelos predictivos, es una técnica estadística para modelar la relación entre variables, esta se puede emplear en muchos campos como la ingeniería, la física, las ciencias, economía, administración, entre otras, también es usada para la minería de datos y para el análisis de datos en general. [31]
Para mostrar las matemáticas involucradas en la tecnica de regresión lineal se va a tener en cuenta el dataset con información relacionada al costo de las casas con respecto a su tamaño, representado en la Tabla **3-1**

Tamaño (Pies cuadrados) X	Precio Y
1500	45
1200	38
1700	48
800	27

Tabla 3-1.: Dataset regresión lineal

En este caso se encuentran la variable de entrada el tamaño de la casa (X) y la variable de salida a predecir, el costo (Y). La ecuación que representa la regresión lineal es la Ecuación 3-1. [24]

$$Y_e = \alpha + \beta * X \tag{3-1}$$

Para predecir el valor de una casa se debe averiguar los valores de las variables α y β, estos valores se calculan con ayuda de la data histórica del dataset, usando el método llamado suma de cuadrados o la diferencia entre Y y Y_e, esta diferencia se representa como la Ecuación 3-2. [24]

$$e = (Y - Y_e) \tag{3-2}$$

El objetivo es minimizar la suma de todos los puntos históricos de X y Y mediante la Ecuación 3-3

$$\sum (Y - Y_e)^2 = \sum (Y - (\alpha + \beta * X))^2 \tag{3-3}$$

Usando el calculo se puede llegar a averiguar los valores de α y β mediante las siguientes ecuaciones 3-4, en donde X_m y Y_m son las medias aritméticas de X y Y. [24]

$$\begin{aligned} \beta &= \frac{\sum (X_i - X_m)(Y_i - Y_m)}{\sum (X_i - X_m)^2} \\ \alpha &= Y_m - \beta * X_m \end{aligned} \tag{3-4}$$

3.1.1. Implementación

Se va a realizar una implementación de la tecnica de regresión lineal, para ello se va a crear un Notebook en Jupyter usando anaconda para ejecutar código en Python.
En el siguiente código se van a crear unos datos aleatorios para poder graficar la tecnica de regresión lineal usando la Ecuación 3-1, se van a crear 100 datos aleatorios con una media de 1.5, desviación estándar de 2.5, una α de 2 y una β de 0.3, dando como resultado la figura **3-2**.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
x = 2.5 * np.random.randn(100) + 1.5
res = 0.5 * np.random.randn(100)
ypred = 2 + 0.3 * x
yact = 2 + 0.3 * x + res

plt.plot(x,ypred)
plt.plot(x,yact,'ro')
plt.title('Actual vs Prediccion')
```

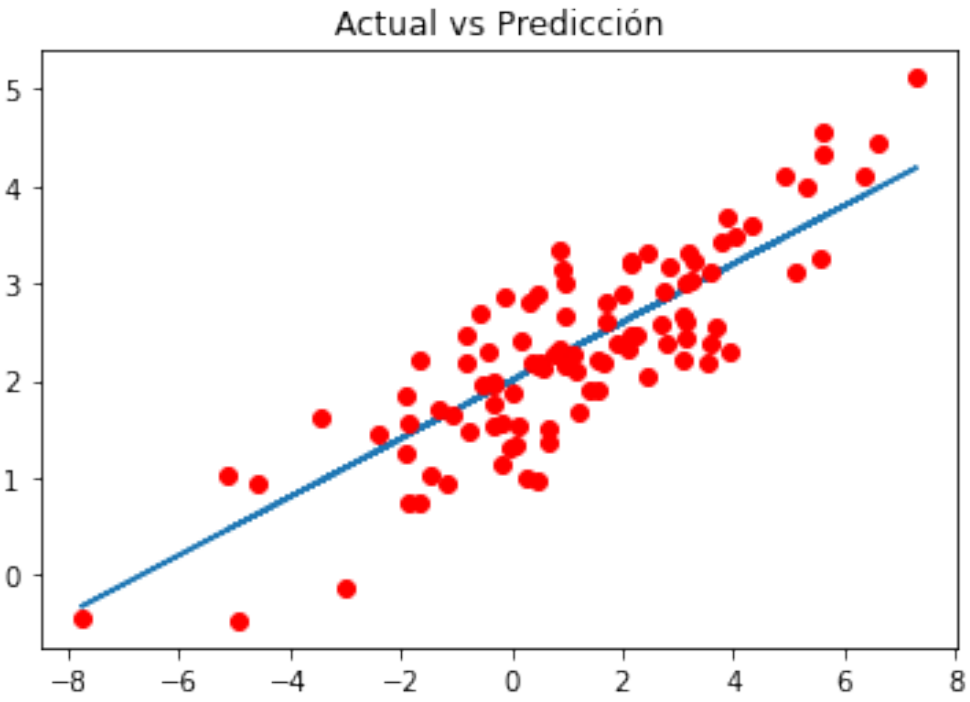


Figura 3-1.: Grafico de regresión lineal

En el grafico los valores aleatorios generados no se aproximan a lo representado por el modelo lineal, en algunos casos un conjunto de datos no es adecuado para modelo de regresión lineal, es posible calcula un coeficiente de relación R^2 el cual es un valor entre 0 y 1, mientras mas próximo a 1 indica una correcta correlación e implica que el modelo es aplicable a ese conjunto de datos, por el contrario si se aleja de 1 implica que el conjunto de datos no es adecuado para este modelo lineal. El coeficiente de correlación se puede calcular mediante la Ecuación 3-5. En la Figura **3-2** se puede apreciar la diferencia entre un dataset con buena y mala correlación.

$$R^2 = \frac{(n+1)\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{\left((n+1)\sum x_i^2 - \left(\sum x_i\right)^2\right)\left((n+1)\sum y_i^2 - \left(\sum y_i\right)^2\right)}} \tag{3-5}$$

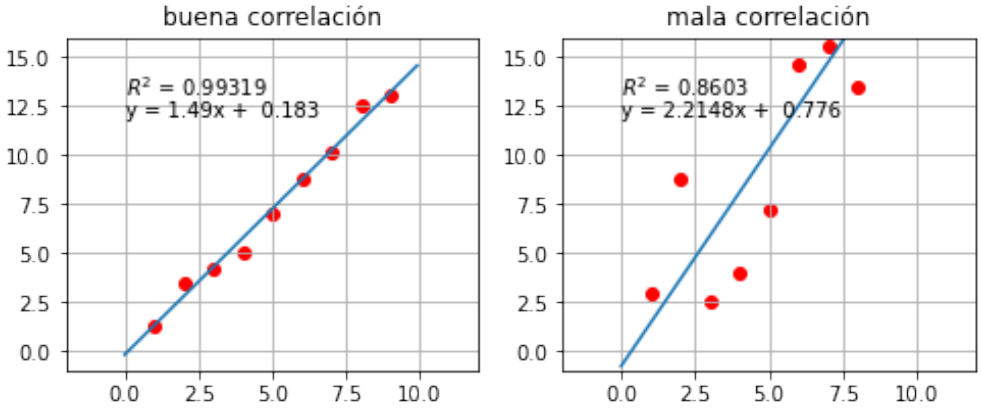


Figura 3-2.: Coeficiente de relación en regresión lineal

3.2. Regresión logística

La regresión logística es una tecnica de análisis predictivo cuyo objetivo es el de comprender una respuesta binaria o proporcional según sea la variable de entrada, con base en unos datos históricos que funcionan como predictores. Como variable de respuesta en la mayoría de ocasiones se obtiene un solo valor binario, 0 para indicar fracaso o ausencia y 1 para el éxito o presencia. [21]
La regresión logística se usa principalmente para definir la probabilidad de que ocurra un evento dado unas variables de entrada, por ejemplo en la Tabla **3-2** se desea predecir si dada una persona compra o no un producto dado su genero. [24]

Genero	Compra
Femenino	Si
Femenino	Si
Femenino	No
Masculino	No
Masculino	Si

Tabla 3-2.: Dataset regresión logística

Para definir la probabilidad condicional de un evento, por ejemplo de que un hombre compre un determinado producto se puede usar la Ecuación 3-6

Genero/Compra	Si	No
Femenino	159	106
Masculino	121	125

Tabla 3-3.: Total compras por genero

$$P = \frac{Numerototaldecomprasporhombres}{Numerototaldehombres}$$
$$P = \frac{121}{246} = 0{,}49 \tag{3-6}$$

Para responder que tan probable es que un evento ocurra se utiliza el concepto de razón de posibilidades (odds ratio) representado por la Ecuación 3-7, el cual indica éxito si el resultado es mayor a 1, mientras mas alto sea el valor mayores oportunidades tendrá de éxito, mientras que si el valor es menor a 1, indica fracaso y mientras mas bajo sea le numero mas probabilidades hay de fracaso. [24]

$$razondecompramasculina = \frac{P_m}{(1-P_m)} \quad razondecompra femenina = \frac{P_f}{(1-P_f)} \tag{3-7}$$

Para calcular las razones de posibilidades de los grupos se utiliza se utiliza la Ecuación 3-8.

$$rmasculino = \frac{\frac{121}{125}}{\frac{159}{106}} = 0{,}64 \qquad rfemenino = \frac{\frac{159}{106}}{\frac{121}{125}} = 1{,}54 \tag{3-8}$$

En este caso por las razones de posibilidades halladas es mas probable que la compra sea exitosa por un hombre que por una mujer, si se da un valor de 1 indica que no hay asociación entre las variables. [24]
Haciendo uso de la regresión logística en donde se calculaba un valor continuo, se adapta a la regresión logística en donde interesa hallar un valor discreto, o o 1, para este caso se plantea la Ecuación 3-9, adaptando la ecuación con lo visto anteriormente y aplicando logaritmo para poder llegar a obtener un valor discreto. [24]

$$\begin{aligned} P &= a + b * X \\ \frac{P}{1-P} &= a + b * X \\ \lg(\frac{P}{1-P}) &= a + b * X \end{aligned} \tag{3-9}$$

Para llegar al valor de P se resuelve mediante la Ecuación 3-10.

$$\begin{aligned} \lg(\frac{P}{1-P}) &= a + b * X \\ \frac{P}{1-P} &= e^{(a+b*X)} \\ P &= \frac{e^{(a+b*X)}}{1+e^{(a+b*X)}} = \frac{1}{1+e^{(a+b*X)}} \end{aligned} \tag{3-10}$$

Para estimar las variables a y b se utiliza el método de máxima verosimilitud (Maximum Likelihood Method) MLE el cual establece que para unos datos multivariables se tienen múltiples variables y n observaciones y la probabilidad de estos se define mediante la Ecuación 3-11. [24]

$$\begin{aligned} L &= P_1 * P_3 * ...P_n(1-P_2)... * (1-P_n) \\ &= \prod_{i=1,y=1}^{n} P_i * \prod_{i=1,y=0}^{n} (1-P_i) \end{aligned} \tag{3-11}$$

Teniendo en cuenta las ecuaciones anteriores de la regresión logística, se obtiene la ecuación MLE Ecuación 3-12. [24]

$$L = \prod_{i=1}^{n} (\frac{e^{(a+b*X_i)} \sum_{i=1}^{n} Y_i}{1 + e^{(a+b*X_i)}}) * (1 - \frac{e^{(a+b*X_i)} \sum_{i=1}^{n} Y_i}{1 + e^{(a+b*X_i)}}) \tag{3-12}$$

La función de MLE se puede definir para cada observación dadas las variables pedictoras, expresando la función en valores de X—P Ecuación 3-14, y usando el método de Newton-Raphson para calcular las raíces, calculando una raíz y actualizado la mediante la siguiente Ecuación 3-13. [24]

$$X_n + 1 = X_n - \frac{f(X)}{f'(X)} \tag{3-13}$$

$$\begin{aligned} L(X|P) &= \prod_{i=1,y=1}^{n} P_{Xi} \prod_{i=1,y=0}^{n} (1 - P_{Xi}) \\ L(X|P) &= \sum_{i=1,y=1}^{n} \lg(P_{Xi}) \sum_{i=1,y=0}^{n} \lg(1 - P_{Xi}) \end{aligned} \tag{3-14}$$

3.2.1. Implementación

Se va a realizar una implementación de un modelo de regresión logística de un conjunto de datos de prestamos para vehículos con el objetivo de predecir el cliente que no pagara el crédito obtenido de [35]. El ejercicio se ejecuta en un notebook de Jupyter, con ayuda de librerías como pandas, numpy, seaborn y sklearn.

```
import pandas as pd
data=pd.read_csv('C:/Niko/EspecializacionSoftware/Proyecto/ejemplo_reg
data.head()
data.dtypes
```

A continuacion, se estudia un ejercicio realizado de regresión logística con el dataset del ejercicio de créditos de vehículo. [36]
En el ejercicio lo primero que se hace es la preparación de la data, limpiando y cambiando los valores como los incorrectos o faltante, cambiando el tipo de dato cuando sea necesario, eliminando o complementando otras variables, a este proceso se le conoce como data wrangling. El data wranglingconsiste en llegar a un conjunto de datos validos para su posterior análisis por ejemplo mediante técnicas de análisis predictivo, este proceso cubre aspectos como el acercamiento al dataset, tener en cuenta el tamaño, la codificación y formato de datos, la calidad de estos datos, la posibilidad de completar los datos faltantes o unirlos, la

documentación de estos datos, entre otros factores que se buscan a la hora de analizar los datos. [16]
Lo primero que se hace es analizar las variables de entrada y posible salida para ver su distribución. [36]

```
import seaborn as sns
import matplotlib.pyplot as plt

# variable a predicir
data["loan_default"].head()
```

```
0    0
1    1
2    0
3    1
4    1
Name: loan_default, dtype: int64
```

Figura 3-3.: Regresión logística - variable a predecir

```
data["disbursed_amount"].describe()
```

```
count    233154.000000
mean      54356.993528
std       12971.314171
min       13320.000000
25%       47145.000000
50%       53803.000000
75%       60413.000000
max      990572.000000
Name: disbursed_amount, dtype: float64
```

Figura 3-4.: Regresión logística - variable dinero prestado

```
sns.catplot(x="loan_default", y="disbursed_amount" , data=data)
```

Como parte del análisis de los datos se ve la relación entre la cantidad prestada, el costo del activo y la edad en la Figura **3-6** y el grupo de edad que presenta mas fraudes en la Figura **3-7**. [36]

```
line1= data.groupby("Age").disbursed_amount.mean()
line2= data.groupby("Age").asset_cost.mean()
```

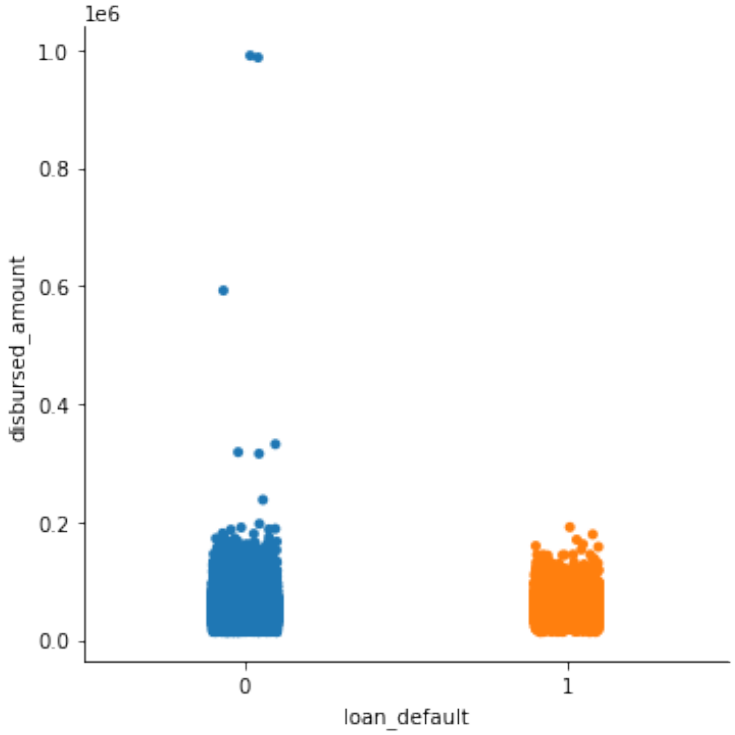


Figura 3-5.: Regresión logística - diagrama caja distribución datos

```
plt.figure(figsize=(10,8))
sns.lineplot(data= line1 , label="disbursed_amount")
sns.lineplot(data=line2 ,label="asset_cost")
plt.title("Cantidad_prestado_y_costo_del_activo_por_grupo_de_edad")
plt.xlabel("Age", size=15)
plt.ylabel("Count", size=15)

pd.crosstab(data.Age, data.loan_default).plot(kind="bar")
```

Después de realizar el análisis de las diferentes variables, se procede a importar los paquetes para realizar la regresión logística, usando sklearn se crea la función, se establece el numero variables a usar en la regresión y se le pasan a la función de regresión logística. [36]

```
from sklearn import datasets
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

lr= LogisticRegression()

len(X) # 37
n= 37
```

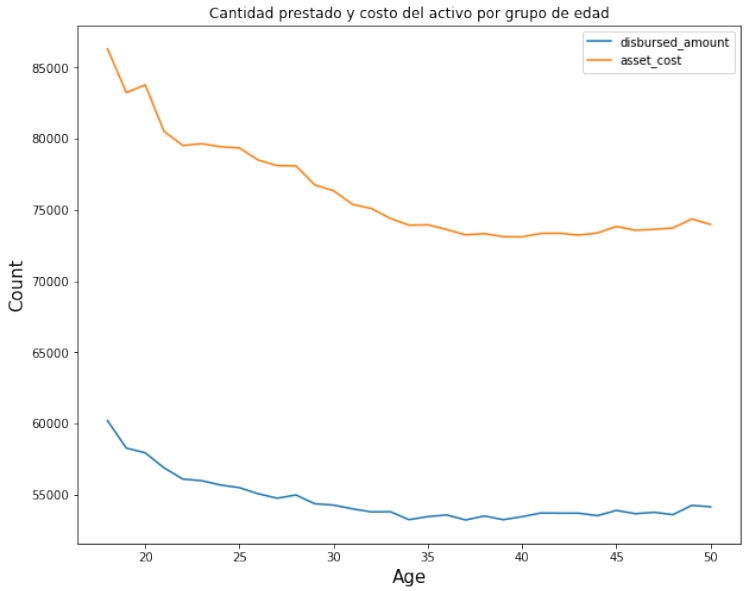


Figura 3-6.: Regresión logística - cantidad prestado, costo activo y edad

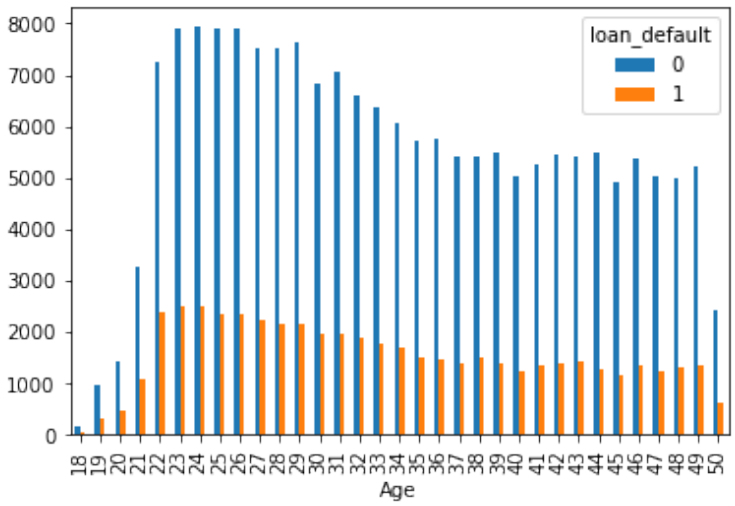


Figura 3-7.: Regresión logística - edad con mas fraudes

```
rfe= RFE(lr ,n)
```

A continuacion, se procede a entrenar el modelo con la librería de sklearn, también se realiza una validación del modelo, obteniendo al final una validación de los datos de entrenamiento en la Figura **3-8**. [36]

```
from sklearn import linear_model
logistic_model= linear_model.LogisticRegression()
logistic_model.fit(X,Y)
```

```
R= logistic_model.score(X,Y)
R*100

output: 78.2897925396628

import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
scores =cross_val_score(linear_model.LogisticRegression(), X,Y, scoring
scores.mean()

output = 0.7828979261589513

lm=linear_model.LogisticRegression()
lm.fit(X_train, Y_train)
probabilidades= lm.predict_proba(X_test)
```

```
from IPython.display import display, Math, Latex
display(Math(r'Y_p=\begin{cases}0& si\ p\leq0.5\\1&si\ p >0.05\end{cases}'))
probabilidades
```

$$Y_p = \begin{cases} 0 & si\ p \leq 0.5 \\ 1 & si\ p > 0.05 \end{cases}$$

```
array([[0.76538255, 0.23461745],
       [0.86449904, 0.13550096],
       [0.90050941, 0.09949059],
       ...,
       [0.77216026, 0.22783974],
       [0.71834022, 0.28165978],
       [0.80568418, 0.19431582]])
```

Figura 3-8.: Regresión logística - validación entrenamiento modelo

3.3. Clustering

El clustering o clasificación es a diferencia de los métodos de regresión lineal y logística, un algoritmo no supervisado, en las técnicas de regresión se precisaban de datos históricos como datos de entrenamiento, la tecnica de clustering no se preocupa por encontrar una variable de salida sino de construir un modelo con las variables predictoras, esta tecnica intenta categorizar las entradas en segmentos que son similares entre ellos, identificar propiedades entre estos grupos o clusters y definir planes o estrategias para cada uno. [24] Dentro de los principales usos para el clustering se encuentra el mercadeo, en especial el digital, el objetivo es encontrar clientes que piensan, se comportan y toman decisiones en lineas similares, según las características de estos grupos mostrar publicaciones, otro uso es el de la taxonomía, encontrar agrupaciones jerárquicas para su clasificación, otro uso es el de encontrar valores a los elementos que faltan en un conjunto de datos. [24]
Para poder encontrar las similitudes entre las variables y poderlas agrupar se usa el principio matemático de la distancia entre puntos, teniendo en cuenta el numero de columnas y de observaciones, para calcular esta distancia se pueden usar diferentes métodos, dentro de estos se encuentra la distancia Euclidiana representada por la Ecuación 3-15, la distancia Manhattan representada por la Ecuación 3-16, la distancia de Minkowski representada por la Ecuación 3-17. [24]

$$D_{ij} = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 + ... + (X_{in} - X_{jn})^2} \tag{3-15}$$

$$D_{ij} = (|X_{i1} - X_{j1}| + |X_{i2} - X_{j2}| + ... + |X_{in} - X_{jn}|) \tag{3-16}$$

$$D_{ij} = ((|X_{i1} - X_{j1}|)^P + (|X_{i2} - X_{j2}|)^P + ... + (|X_{in} - X_{jn}|)^P)\frac{1}{P} \tag{3-17}$$

Una vez calculada la distancia entre dos puntos se puede almacenar esa distancia en una matriz, la matriz representa la distancia entre esos 2 puntos que representan 2 observaciones en el dataset, una matriz de distancias debe tener las siguientes propiedades: [24]

- Todos los elementos de la diagonal de la matriz debn ser 0 ya que son la distancia entre el mismo punto $D_{ij} = 0$
- Para un conjunto de datos de N observaciones se tiene una matriz de NxN
- La matriz es simétrica, la distancia entre el punto 1 y 2 es la misma que del punto 2 al 1 $D_{ij} = D_{ji}$

La Tabla **3-4** representa un ejemplo de una matriz de distancia.

	1	2	3	4
1	0.00	0.67	1.01	0.73
2	0.67	0.00	0.55	0.06
3	1.01	0.55	0.00	0.81
4	0.73	0.06	0.81	0.00

Tabla 3-4.: Matriz de distancia

3.3.1. K-means clustering

El agrupamiento por el algoritmo k-means es un algoritmo de clustering con el cual se agrupan las observaciones en grupos jerárquicos, este algoritmo tiene un enfoque incremental para la agrupación, agregando de manera dinámica un centro de cluster a través de una procedimiento de búsqueda determinista que consta de N ejecuciones de las medias del conjunto de observaciones. [25]
La ejecución de este algoritmo sigue los siguientes pasos: [24]

- Decidir el numero de grupos y asignar el centroide para cada grupo, este numero depende del contexto del ejercicio, las observaciones elegidas al azar pueden servir como centroides del grupo.
- Calcular la distancia de cada observación en cada grupo, asignando la observacion al cluster con la distancia menor
- Re calcular el centroide del cluster usando la media de la observación en el cluster mediante la Ecuación 3-18 en donde cies el numero de observaciones en el cluster y Xm es el vector de observación cuyo la longitud es igual al número de columnas de la observación.
- Repetir los pasos a partir del paso 2
- Detenerse si ninguna observación se reasigna de un cluster a otro, esto indica que la distancia de a un centroide de un grupo no pude ser reducida más.

$$vi = \frac{1}{ci} \sum_{m=1}^{ci} Xm \tag{3-18}$$

El objetivo de este algoritmo es lograr una configuración de clusters y observaciones de modo que se minimice la función de error de j, representado en la Ecuación 3-19

$$J = \sum_{i=1}^{c} \sum_{j=1}^{ci} (Xj - Vi)^2 \tag{3-19}$$

3.3.2. Implementación

Se va a realizar una implementación del algoritmo de clustering k-mean, mediante un dataset de clientes de un centro comercial, se ejecuta en Jupyter notebook haciendo uso de librerías en python con ayuda de las librerías importadas en el siguiente código, este ejemplo fue tomado de kaggle [27].

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_c
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
import plotly.graph_objs as go
from sklearn.cluster import KMeans
import warnings
import os
```

```
df = pd.read_csv('C:/Niko/EspecializacionSoftware/Proyecto/ejemplo_clustering/Mall_Customers.csv')
df.head()
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Figura 3-9.: Dataset prueba k-means clustering

Para realizar el agrupamiento de la data primero se analiza la dispersión de este, para esto se tienen en cuenta las variables de edad, ingresos anuales y genero en la Figura **3-10** y las variables de ingresos anuales, puntaje de gasto y genero en la Figura **3-11**.

```
plt.figure(1 , figsize = (15 , 6))
for gender in ['Male' , 'Female']:
plt.scatter(x = 'Age' , y = 'Annual Income (k$)' , data = df[df['Ge
s = 200 , alpha = 0.5 , label = gender)
plt.xlabel('Edad'), plt.ylabel('Ingresos anuales (k$)')
plt.title('Edad vs Ingresos anuales por Genero')
plt.legend()
plt.show()

plt.figure(1 , figsize = (15 , 6))
```



Figura 3-10.: Grafico edad vs ingresos por genero

```
for gender in ['Male', 'Female']:
plt.scatter(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',
data = df[df['Gender'] == gender] ,s = 200 , alpha = 0.5 , label = gend
plt.xlabel('Ingresos anuales (k$)'), plt.ylabel('Puntaje de gasto (1-10
plt.title('Ingresos anuales vs Puntaje de gasto por Genero')
plt.legend()
plt.show()
```

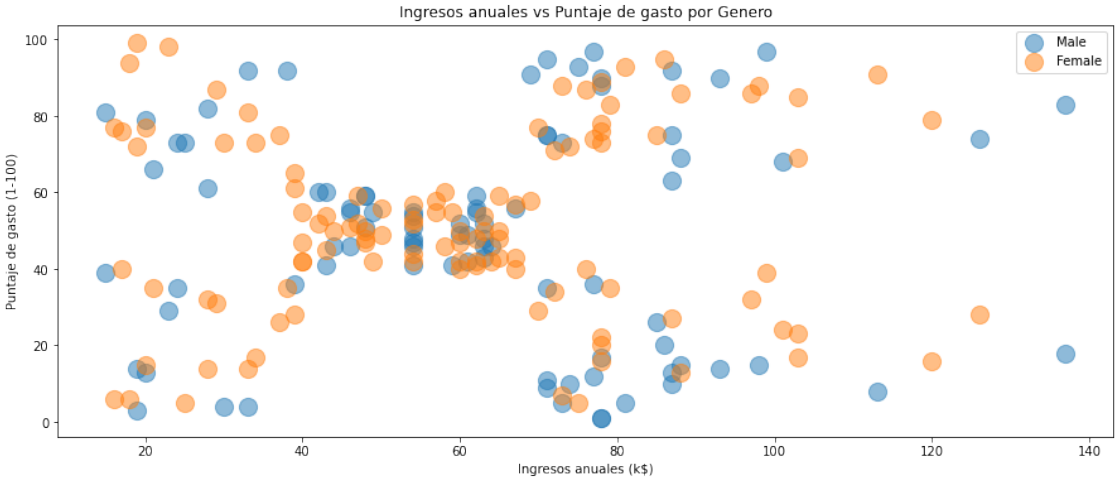


Figura 3-11.: Grafico ingresos vs puntaje por genero

Para empezar a utilizar el método de k.means se hace uso de KMeans de sklearn, en el siguiente código se realiza la segmentación por edad y puntaje de gasto.

```
X1 = df[['Age' , 'Spending Score (1-100)']].iloc[: , :].values
```

```
inertia = []
for n in range(1 , 11):
algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10 ,
tol=0.0001,  random_state= 111  , algorithm='elkan') )
algorithm.fit(X1)
inertia.append(algorithm.inertia_)
```

La seleccion del numero de clusters se realiza en el método de Inertia de la misma libreria de KMeans, en la Figura **3-12** se observa este proceso.

```
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Numbero de clusters') , plt.ylabel('Inertia')
plt.show())
```

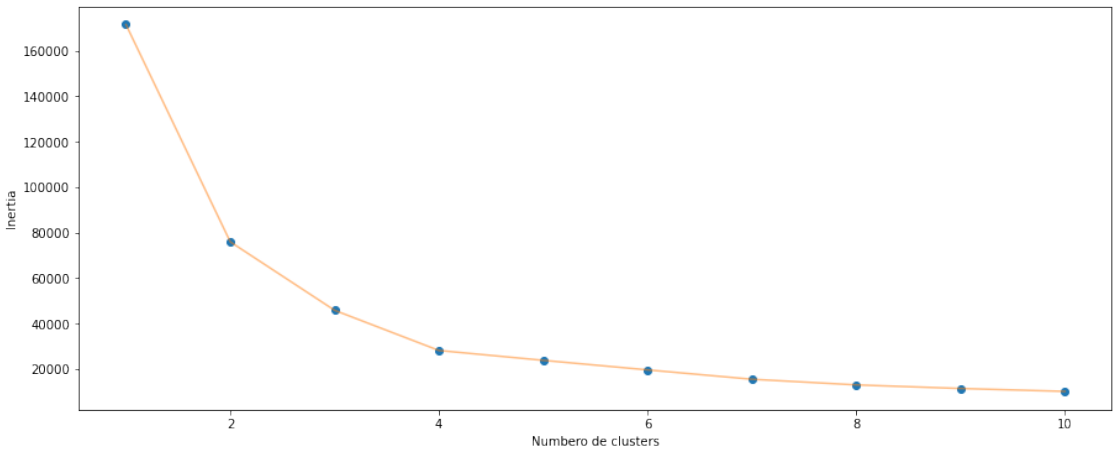


Figura 3-12.: Grafico numero de clusters por Inertia

Se realizan las operaciones para establecer los centroides y buscar los mínimos y máximos del data set.

```
algorithm = (KMeans(n_clusters = 4 ,init='k-means++', n_init = 10 ,
tol=0.0001,  random_state= 111  , algorithm='elkan') )
algorithm.fit(X1)
labels1 = algorithm.labels_
centroids1 = algorithm.cluster_centers_

h = 0.02
x_min, x_max = X1[:, 0].min() - 1, X1[:, 0].max() + 1
y_min, y_max = X1[:, 1].min() - 1, X1[:, 1].max() + 1
```

```
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max
Z = algorithm.predict(np.c_[xx.ravel(), yy.ravel()])
```

Por ultimo se grafica la dispersión por los grupos segmentados gracias al método de k-means Figura **3-13**.

```
plt.figure(1 , figsize = (15 , 7) )
plt.clf()
Z = Z.reshape(xx.shape)
plt.imshow(Z , interpolation='nearest',
extent=(xx.min(), xx.max(), yy.min(), yy.max()),
cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')

plt.scatter( x = 'Age' ,y = 'Spending Score (1-100)' , data = df , c =
s = 200 )
plt.scatter(x = centroids1[: , 0] , y = centroids1[: , 1] , s = 300 ,
plt.ylabel('Puntaje de gasto (1-100)') , plt.xlabel('Edad')
plt.show()
```

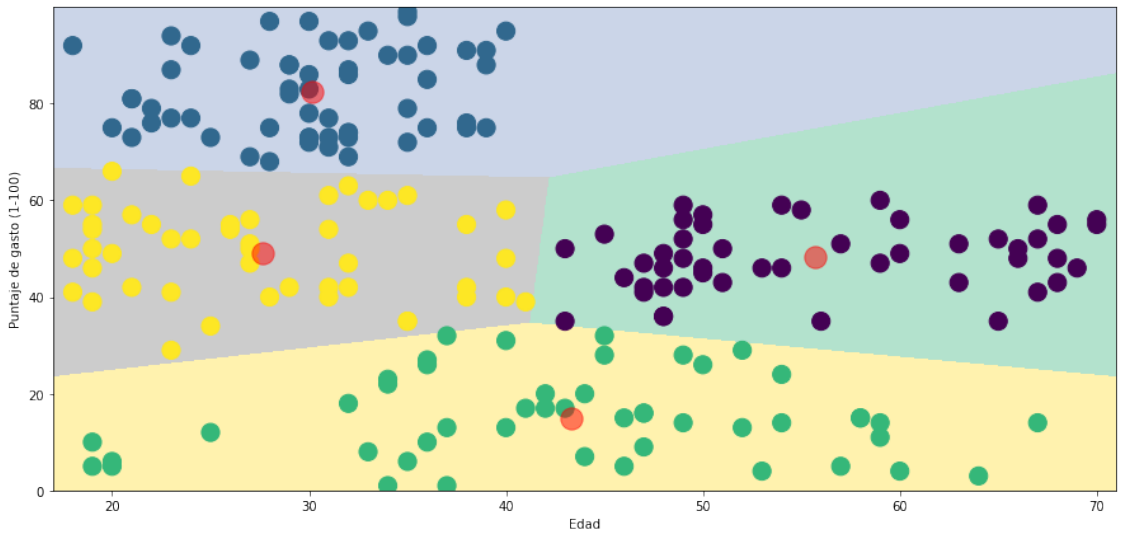


Figura 3-13.: Grafico segmentación k-means

3.4. Arboles de decisión

Un árbol de decisión es un modelo basado en el principio de divide y vencerás, en donde el conjunto de datos de entrenamiento esta particionado o se divide de forma recursiva en regiones rectangulares no superpuestas, estas regiones o cajas multi dimensionales están asociadas con una regla de decisión responsable de la clasificación de la región. [18]
Un árbol de decisiones es un algoritmo de clasificación supervisado en el que la variable objetivo puede ser discreta o categórica, del mismo modo las variables predictoras. Un árbol de decisión puede representarse como un conjunto de reglas si entonces para un problema de clasificación donde las variables objetivo pueden ser discretas o categóricas, estas reglas son representadas por un árbol. Un árbol de decisiones es una herramienta de toma de decisiones eficaz ya que tiene un grafico de salida el cual es muy fácil de entender e implementar en comparación con la salida de otros modelos predictivos. La estructura de un árbol consiste de nodos, ramas y hojas, los nodos son los puntos de donde salen una o mas ramas, un nodo que no genera ninguna rama es un hoja, como se aprecia en la Figura **3-14**. [24]

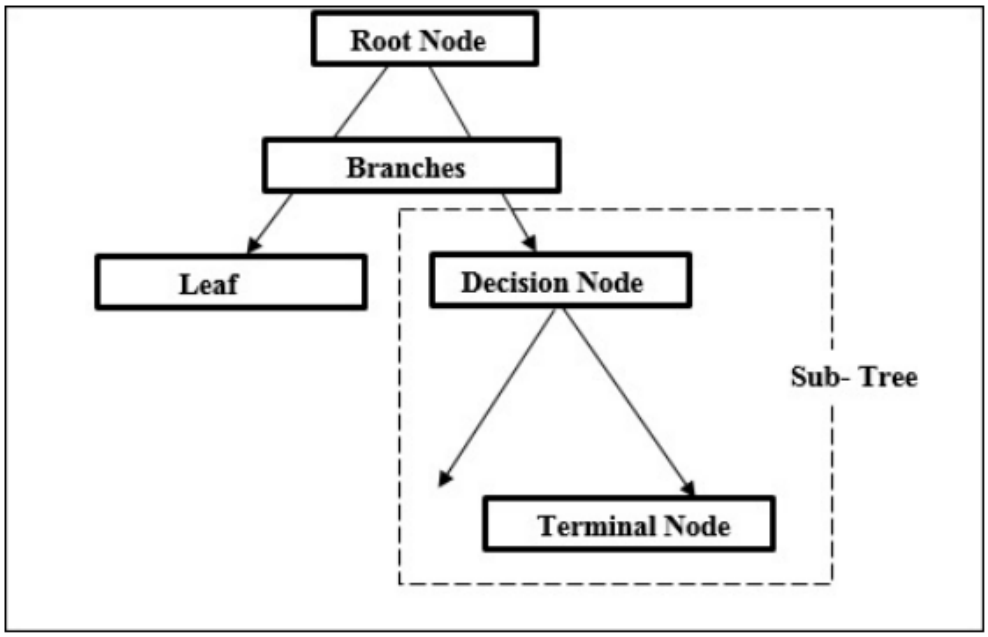


Figura 3-14.: Estructura de un árbol [24]

El objetivo de un algoritmo de árbol de decisiones es el de identificar una variable y una clasificación en la que se pueda dar una distribución mas homogénea con respecto a la variable objetivo, esta distribución significa que los valores similares de la variable objetivo se agrupan para que se pueda tomar una decisión. [24]
Para entender el funcionamiento del algoritmo de árbol de decisiones es importante destacar el concepto de homogeneidad, mediante el cual se busca encontrar un parámetro que resulte en una mejor distribución homogénea de la variable de destino dentro de las categorías.

Otro concepto importante es el de la entropía, tomada de la teoría de la información, la entropía en este contexto establece que los nodos mas homogéneos o puros requieren menos información para ser representados, la reducción de la entropía no deseable se puede medir con ganancia de la información, para el caso de un árbol de decisión, los nodos cuya adición da como resultado una ganancia de información deben agregarse a la configuración. Esta medida de entropia puede definirse como lo muestra la Ecuación 3-20. [24]

$$Entropia(S) = \sum -p_i log_2 p_i \tag{3-20}$$

Después de calcular la entropía del sistema, esta se reduce a medida que se introducen variables como nodos, esta reducción se refleja como una ganancia de información, esta se puede calcular para una variable determinada mediante la Formula 3-21. [24]

$$GananciaInformacion(S, V) = S_v - \sum (\frac{V_c}{V}).Entropia(V_c) \tag{3-21}$$

Uno de los métodos usados para cuando la variable de destino es binaria es el método Gini index, este utiliza el algoritmo CART el cual aborda el problema de clasificación y regresión construyendo un árbol de decisión binario de acuerdo con alguna regla de división basada en las variables predictoras. De esta forma, el espacio de las variables predictoras se particiona de forma recursiva de forma binaria. La partición está destinada a aumentar la homogeneidad dentro de los nodos, donde la homogeneidad está determinada por la variable de respuesta en el problema [12]. El metodo Gini se define como la suma del cuadrado de las proporciones de las categorías de la variable objetivo para la categoría particular de la variable de nodo. El calculo de Gini index se da mediante las Formulas 3-22. [24]

$$\begin{aligned} Gini_{C1} &= (True_{C1})^2 + (False_{C1})^2 \\ Gini_{C2} &= (True_{c2})^2 + (False_{C2})^2 \\ Gini_{Cn} &= (True_{Cn})^2 + (False_{Cn})^2 \\ GiniIndex &= \sum_{i=1}^{n} (\frac{ObservacionesC_i}{ObservacionesTotales} * Gini_{Cn}) \end{aligned} \tag{3-22}$$

Un ejemplo para los datos de la Tabla **3-5** se aprecia en la Formula 3-23.

$$\begin{aligned} Gini_{C1} &= (0{,}2)^2 + (0{,}8)^2 = 0{,}68 \\ Gini_{C2} &= (0{,}6)^2 + (0{,}4)^2 = 0{,}52 \\ GiniIndex &= (\frac{10}{20}) * 0{,}68 + (\frac{10}{20}) * 0{,}52 = 0{,}60 \end{aligned} \tag{3-23}$$

Categoría	Si	No	Total
C1	2	8	10
C2	6	4	10
Total	8	12	20

Tabla 3-5.: Valores ejemplo Gini index

Un aspecto importante en los arboles de decisión es la poda de árbol, eliminar ramas o nodos que no proporcionan mucho poder de clasificación, esto es necesario para cuando un arbol de decisiones sobre aprende un conjunto de datos de entrenamiento que no funcionan tan bien con otros conjuntos de datos similares.

3.4.1. Implementación

Para la implementación de la tecnica de arboles de decisiones, se utilizo un conjunto de datos de evaluación de automóviles para predecir la seguridad de un automóvil, utilizando los criterios de Gini index y de entropía. El ejercicio se ejecuta en Jupyter Notebook haciendo uso de librerías en Python, este ejemplo fue tomado de kaggle [3]. Se importan las librerías necesarias y el dataset.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_c
import matplotlib.pyplot as plt # data visualization
import seaborn as sns # statistical data visualization
%matplotlib inline

data = 'C:/Niko/EspecializacionSoftware/Proyecto/ejemplo_arboles/ca
df = pd.read_csv(data, header=None)
```

Lo primero que se hace es analizar los datos, se cambian los nombres de las columnas y se analizan los tipos de datos de las variables y se buscan posibles datos nulos o incorrectos, ver Figura **3-15**.
Se declara un vector de características y una variable de destino, se seoaran los datos en conjutos de prueba y entrenamiento.

```
X = df.drop(['class'], axis=1)
y = df['class']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
```

	0	1	2	3	4	5	6
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

```
col_names = ['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class']
df.columns = col_names

col_names
```

```
['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class']
```

```
df['class'].value_counts()
```

```
unacc    1210
acc       384
good       69
vgood      65
Name: class, dtype: int64
```

```
df.isnull().sum()
```

```
buying      0
maint       0
doors       0
persons     0
lug_boot    0
safety      0
class       0
dtype: int64
```

Figura 3-15.: Arboles de decisión - análisis datos

Se codifican las variables categóricas para poder dejar los latos listos para su posterior análisis, dando como resultado la Figura **3-16**.

```
import category_encoders as ce

encoder = ce.OrdinalEncoder(cols=['buying', 'maint', 'doors', 'persons'
X_train = encoder.fit_transform(X_train)
X_test = encoder.transform(X_test)
```

Ahora se va a implementar el método de Gini index, esto mediante la librería sklearn.

```
from sklearn.tree import DecisionTreeClassifier

clf_gini = DecisionTreeClassifier(criterion='gini', max_depth=3, randon
clf_gini.fit(X_train, y_train)
```

```
X_train.head()
```

	buying	maint	doors	persons	lug_boot	safety
48	1	1	1	1	1	1
468	2	1	1	2	2	1
155	1	2	1	1	2	2
1721	3	3	2	1	2	2
1208	4	3	3	1	2	2

```
X_test.head()
```

	buying	maint	doors	persons	lug_boot	safety
599	2	2	4	3	1	2
1201	4	3	3	2	1	3
628	2	2	2	3	3	3
1498	3	2	2	2	1	3
1263	4	3	4	1	1	1

Figura 3-16.: Arboles de decisión - variables codificadas

```
output: DecisionTreeClassifier(max_depth=3, random_state=0)
```

Se ejecuta la predicción de los resultados del conjunto de pruebas con el método de Gini index.

```
y_pred_gini = clf_gini.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
```

```
print('Model accuracy score with criterion gini index: {0:0.4f}'. f
```

```
output: Model accuracy score with criterion gini index: 0.8021
```

Ahora se visualizan los arboles de decisión con el método de Gini index dando como resultado la Figura **3-17**.

```
plt.figure(figsize=(12,8))
```

```
from sklearn import tree
```

```
tree.plot_tree(clf_gini.fit(X_train, y_train))
```

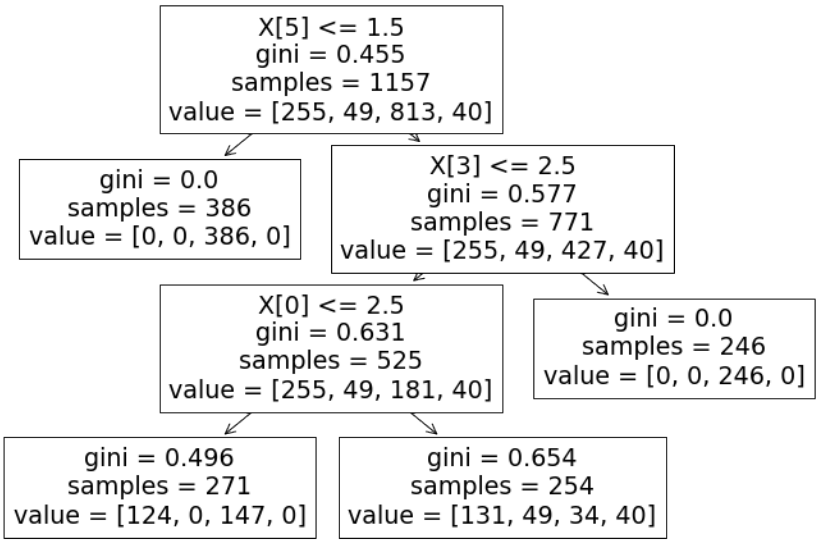


Figura 3-17.: Arboles de decisión - Grafico Gini index

Ahora se va a utilizar el criterio de entropía para el mismo conjunto de datos

```
clf_en = DecisionTreeClassifier(criterion='entropy', max_depth=3, rando
clf_en.fit(X_train, y_train)
```

```
output: DecisionTreeClassifier(criterion='entropy', max_depth=3, randon
```

Se predice el resultado del conjunto de datos de prueba con el criterio de entropía

```
y_pred_en = clf_en.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
```

```
print('Model accuracy score with criterion entropy: {0:0.4f}'. format(a
```

```
output: Model accuracy score with criterion entropy: 0.8021
```

Se visualizan los arboles de decisión con el método de entropía dando como resultado la Figura **3-18**.

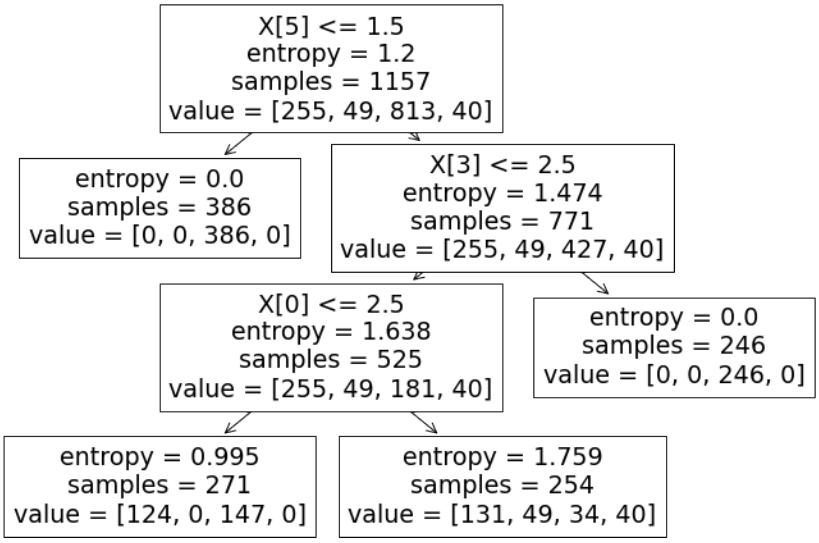


Figura 3-18.: Arboles de decisión - Grafico entropía

4. ARQUITECTURA EMPRESARIAL

En este capitulo se diseña una prepuesta de arquitectura empresarial para la solución de software ofrecida mediante una aplicación móvil para el control y administración de las finanzas personales, integrando un modelo de análisis predictivo de datos. Parte del diseño de esta prepuesta se basa en la creación de una empresa de emprendimiento tecnológico que de soporte a los procesos de negocio que impulsan la creación de la aplicación.
Para la arquitectura propuesta se emplea el método ADM de TOGAF expuesto en este capitulo y el lenguaje de modelado utilizado es Archimate, para el diseño de los diagramas de las diferentes capas de la arquitectura empresarial.

4.1. Organización

4.1.1. Introducción

Debido a que la aplicación que se pretende poner en el mercado requiere de un grupo de trabajo el cual de soporte a los diferentes aspectos que esta conlleva, se ve la necesidad de crear una empresa la cual es la base y figura legal que respaldara y dará seguridad a nuestro mercado en el momento de adquirir dicha aplicación.

4.1.2. Empresa

El modelo de empresa que se crea es un startup modelo utilizado en las compañías que se encuentran especializadas en el campo de la tecnología, que desean generar innovación y realizan la integración de datos web que permiten generar servicios que beneficien a nuestros usuarios en este caso en el ámbito financiero familiar.

4.1.3. Misión

Facilitar el manejo de los temas cotidianos por medio de aplicaciones que permitan la evaluación de varias situaciones y generen resultados óptimos para los hogares en Colombia. Proporcionar a la comunidad en general herramientas seguras, rápidas y eficaces que ayuden a la estimación de los movimientos financieros y así mismo a la toma de decisiones a corto, mediano y largo plazo.

4.1.4. Visión

En tres años queremos ser una empresa líder en la creación de aplicaciones que acerquen la tecnología a todos los habitantes del territorio nacional, por medio de la demostración de utilidad de las aplicaciones en el día a día.

4.1.5. Objetivos organizacionales

• Desarrollar aplicaciones móviles de fácil uso y con una correcta funcionalidad para la resolución de problemas cotidianos.

• Crear alianzas estratégicas junto a proveedores que proporcionen servicios de buena calidad y precio, para maximizar beneficios a clientes internos y externos.

• Potenciar el área de Talento Humano con profesionales idóneos.

• Manejar excelentes relaciones comerciales y emplear oportunas estrategias de marketing.

4.1.6. Procesos

Análisis de las necesidades de las familias colombianas y que puedan ser mejoradas o resueltas por medio de aplicaciones, luego se pasa al proceso de planteamiento de la aplicación en el cual se decide el lenguaje que se manejara, así como las herramientas que esta tendrá para que se ha fácil que el usuario se desplace entre los diferentes links; pasamos a la parte de desarrollo como tal y se realizan las pruebas de funcionalidad antes de ser sacada al mercado. Luego de superar todas las pruebas con excelentes resultados se procede al lanzamiento de la aplicación al mercado.

4.1.7. Servicios

A través de la aplicación móvil se brinda el servicio de análisis predictivo para el control del comportamiento financiero mediante gestión, análisis y recomendaciones de gastos personales

4.1.8. Productos

Aplicación móvil de fácil acceso y manejo disponible para todos los hogares colombianos.

4.1.9. Requerimientos organizacionales

Entorno
• La aplicación móvil debe ser diseñada y construida con los mayores niveles de flexibilidad en cuanto a la parametrización de los tipos de datos, debe ser además fácil de instalar en todas las plataformas de hardware y software de base requeridas, así como permitir su instalación en diferentes tamaños de configuración.
Operacionales
• Permitir denegar el acceso al sistema cuando no se aplique un mecanismo de autenticación aceptado y atribuido al perfil del usuario. • Permitir generar informes o reportes junto a una organización por fechas o secuencias cronológicas. • Permitir contar con procedimientos automáticos para copias de seguridad y restauración encaminados a realizar copias periódicas de seguridad de todos elementos.
Desarrollo
• Requiere lenguaje de programación Python • Plantear un diseño en el cual se base el desarrollo de la aplicación. • Ejecutar pruebas que avalen el correcto funcionamiento de la aplicación.

4.1.10. Estructura orgánica

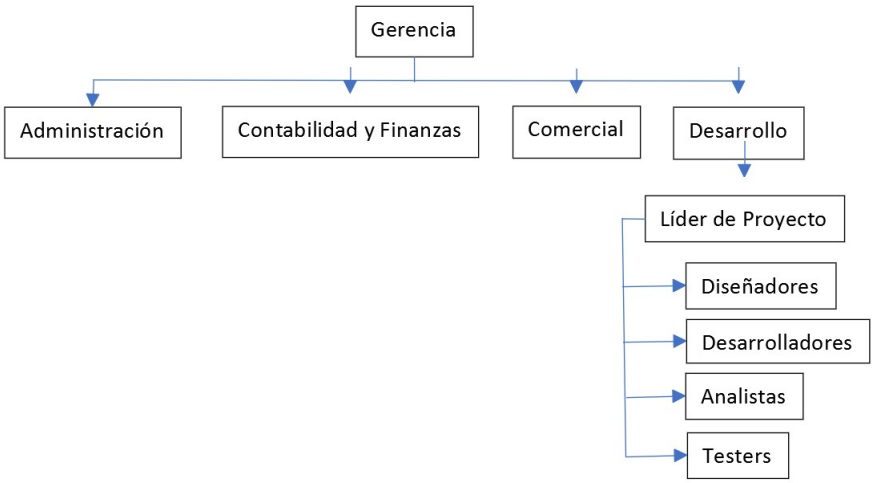


Figura 4-1.: Estructura orgánica

4.2. Lenguaje y el método

4.2.1. ADM

El Método de Desarrollo de la Arquitectura, conocido por sus siglas en inglés ADM hace parte de la metodología de implementación del TOGAF, este último es un marco de referencia para la definición de la arquitectura empresarial y está orientado a promover una disciplina para coordinar los componentes de una organización, permitiendo trazar estrategias de evolución de la arquitectura actual hacia una arquitectura futura que proporcione una ventaja competitiva, flexibilidad y agilidad a la organización. El ADM considera las siguientes etapas:

- Preliminar: iniciar el proceso de adopción del ADM al interior de la organización.
- Visión de arquitectura: se define el alcance y la estrategia.
- Arquitectura de negocio: obtener la arquitectura de negocio y sus metas para alinear las TI al negocio.
- Arquitectura de sistemas de información: determinar las arquitecturas particulares para datos y aplicaciones.
- Arquitectura tecnológica: identificar la arquitectura de la infraestructura tecnológica y de comunicaciones.
- Oportunidades y soluciones: determinar y evaluar los proyectos de implementación de arquitectura.
- Planificación de la migración: analizar costos, beneficios y riesgos; desarrollar una lista priorizada de proyectos.
- Gobierno de la implementación: mantener el control y aseguramiento de las implementaciones alineadas a la arquitectura.
- Gestión de cambios de la arquitectura: monitorear y evaluar las arquitecturas existentes para determinar cuándo iniciar un nuevo ciclo.

Dentro de los beneficios para la organización se destacan el obtener una visión integral del modelo del negocio, promoviendo la flexibilidad y adaptabilidad para aprovechar oportunidades; aumentar la agilidad de la organización y el desarrollo de nuevas innovaciones y capacidades; visibilidad del uso de los recursos y mejorar la toma de decisiones de negocio. En cuanto a los beneficios para la TI, permite visibilidad dentro de la organización; optimiza las inversiones en TI, así como el uso más efectivo de sus recursos; permite plataformas que garanticen la operación eficiente de los servicios de información, disminución de costos de mantenimiento y soporte; portabilidad de aplicaciones y un entorno más flexible para la gestión de cambios. [33]



Figura 4-2.: ADM

4.2.2. Archimate

Es un estándar (metamodelo y lenguaje de modelado) creado por OpenGroup, ideado para crear modelos de arquitectura empresarial, que son representaciones integradas de todos los aspectos de interés en una organización, desde estrategia hasta hardware, pasando por negocio y sistemas. Las arquitecturas empresariales se han convertido en una herramienta clave para apoyar a los directores de TI a la hora de gestionar la gran complejidad que deben enfrentar. Las ventajas que se obtienen son: tener un lenguaje internacional, simple y limpio, fácil de combinar con métodos de desarrollo de la arquitectura y soportado por múltiples AE. Archimate permite representar gráficamente y describir la arquitectura en diversas capas, está compuesto por 4 dimensiones de modelado, 6 capas de servicio en el modelo completo; no obstante, puede representarse con 3 capas de servicios y 3 dimensiones de modelado; en las cuales se encuentran: [2]

- Strategy: representa un activo de propiedad o controlado por un individuo u organización.
- Business: ofrece productos y servicios a los clientes externos, realizados en la organi-

zación por procesos de negocio y llevados a cabo por actores o roles del negocio.

- Application: soporta la capa de negocio (superior) con servicios de aplicación que son realizados por componentes de la aplicación (sistemas software).
- Technology: ofrece servicios de infraestructura (procesamiento, almacenamiento, comunicación) necesarios para ejecutar las aplicaciones.
- Physical: muestra los elementos físicos que son la base de technology.
- Implementation: muestra los elementos para la implementaciones y migración.
- Passive Structure: elementos (estáticos) que son los objetos sobre los que se realiza el comportamiento.
- Behavior: elementos (dinámicos) que son unidades de actividad realizadas por entidades estructurales activas.
- Active Structure: Entidades (estáticas) que son capaces de tener comportamiento.
- Motivation: elementos de motivación se utilizan para modelar las motivaciones o razones, que guían el diseño o el cambio de una arquitectura empresarial.

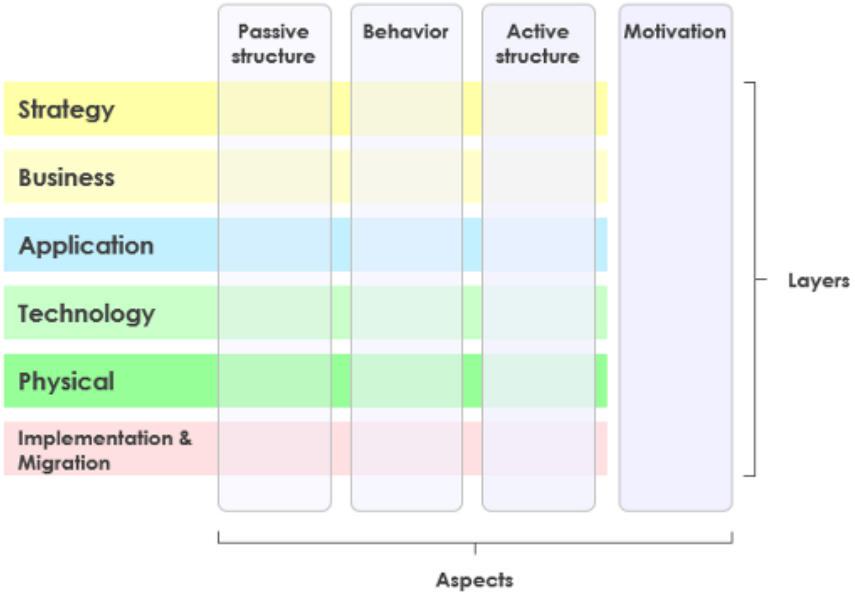


Figura 4-3.: ADM

4.3. Capa de motivación

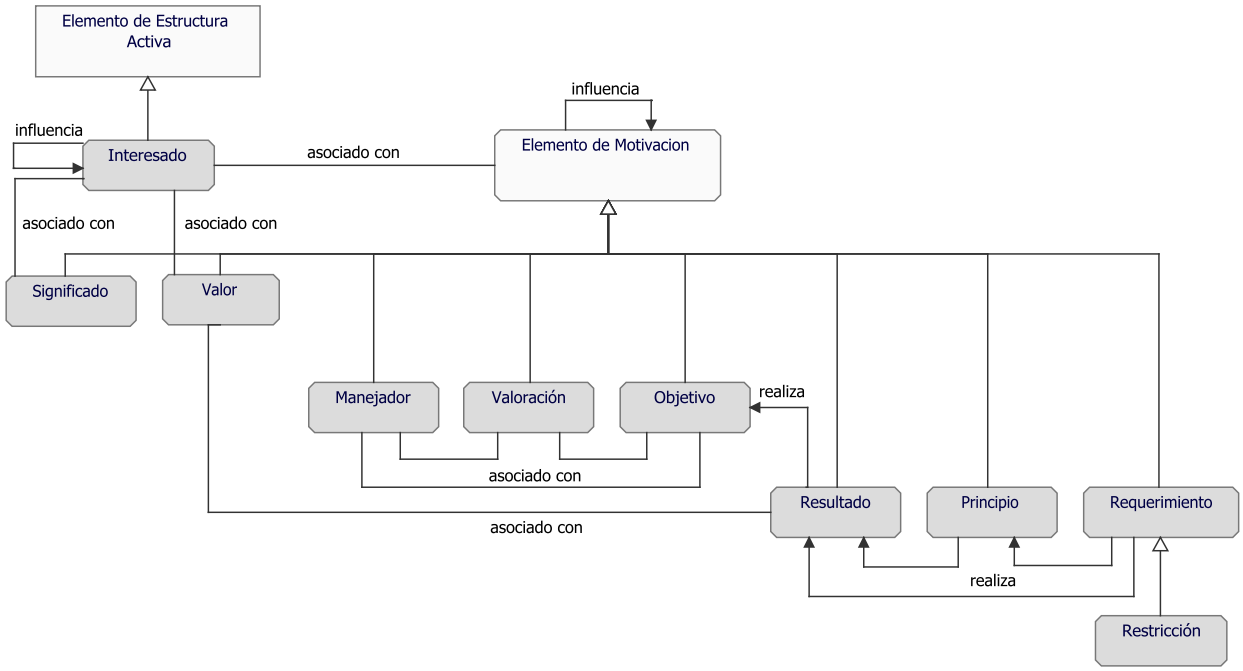


Figura 4-4.: Metamodelo de Motivación

4.3.1. Introducción

En esta capa se destacan los diferentes aspectos motivacionales de la organización, se pretende identificar las entidades como objetivos y principios, como se relacionan y hacen parte fundamental de la estructura organizacional, dentro de esta estructura se identifican relaciones a nivel externo como los son principios e interesados y a nivel interno como lo son objetivos y requerimientos, así como los diferentes roles que interactúan en cada escenario estableciendo una comunicación entre los diferentes elementos. [5]

4.3.2. Punto de vista de interesados

Mediante este punto de vista se modelan las partes interesadas, impulsores internos y externos del cambio, se describen los vínculos de los objetivos iniciales de alto nivel, los cuales forman base para el proceso de ingeniería de requisitos. Los elementos para este punto de vista se observan en la figura **4-5**, el caso modelado propuesto se representa en la figura **4-6**

Modelo de interesados

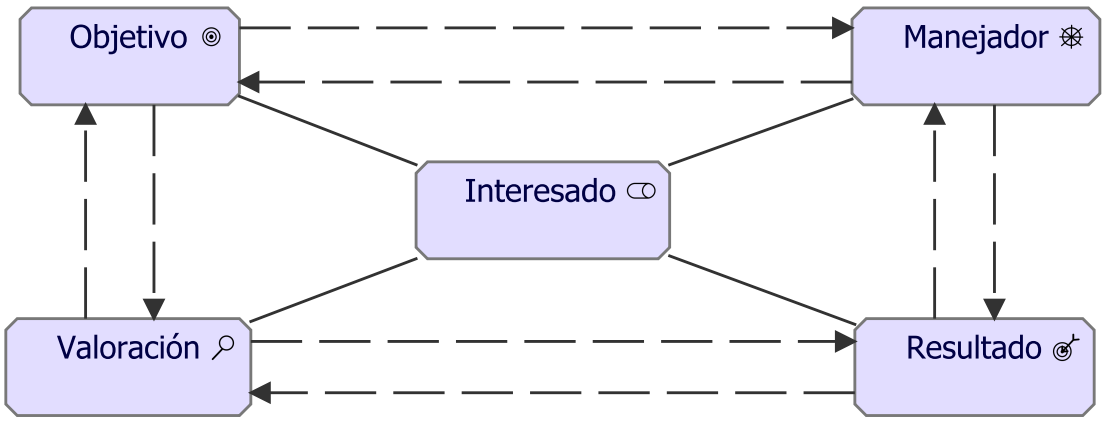


Figura 4-5.: Interesado

Caso de stakeholder

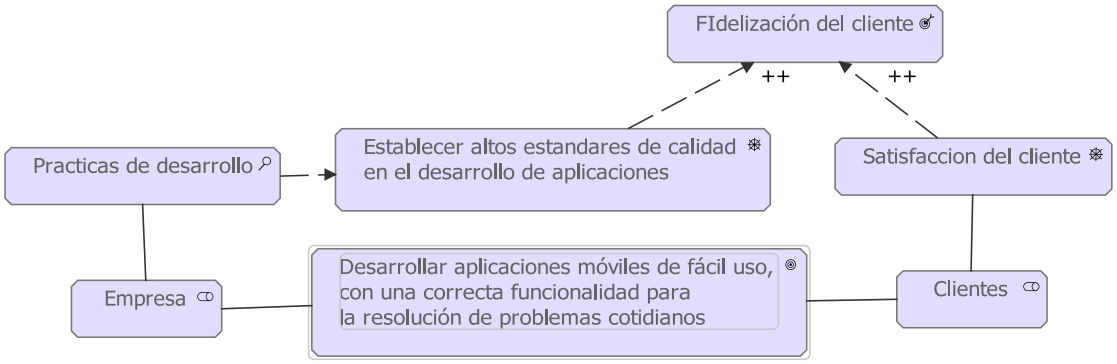


Figura 4-6.: Caso Interesado

4.3.3. Punto de vista de realización de objetivos

En este punto de vista se permite modela el refinamiento de metas de alto nivel en metas mas concretas, el refinamiento de objetivos concretos en requisitos y restricciones que describen las propiedades que se necesita para llevar a cabo estas metas. Los elementos para este punto de vista se observan en la figura **4-7**, el caso modelado propuesto se representa en la figura **4-8**

Modelo de realización de objetivos

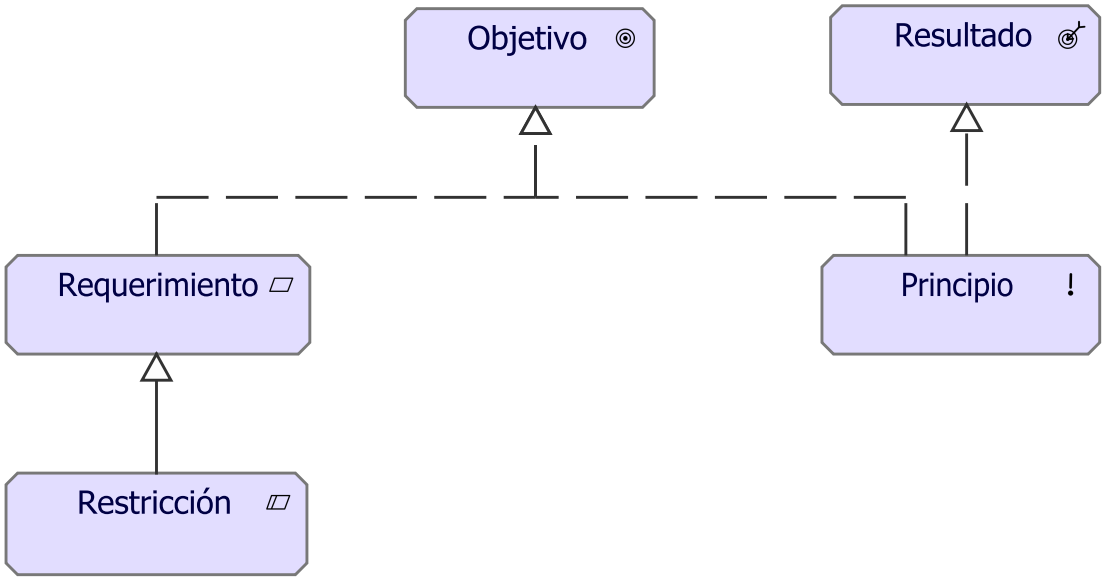


Figura 4-7.: Realización de Objetivos

Caso de realización de objetivos

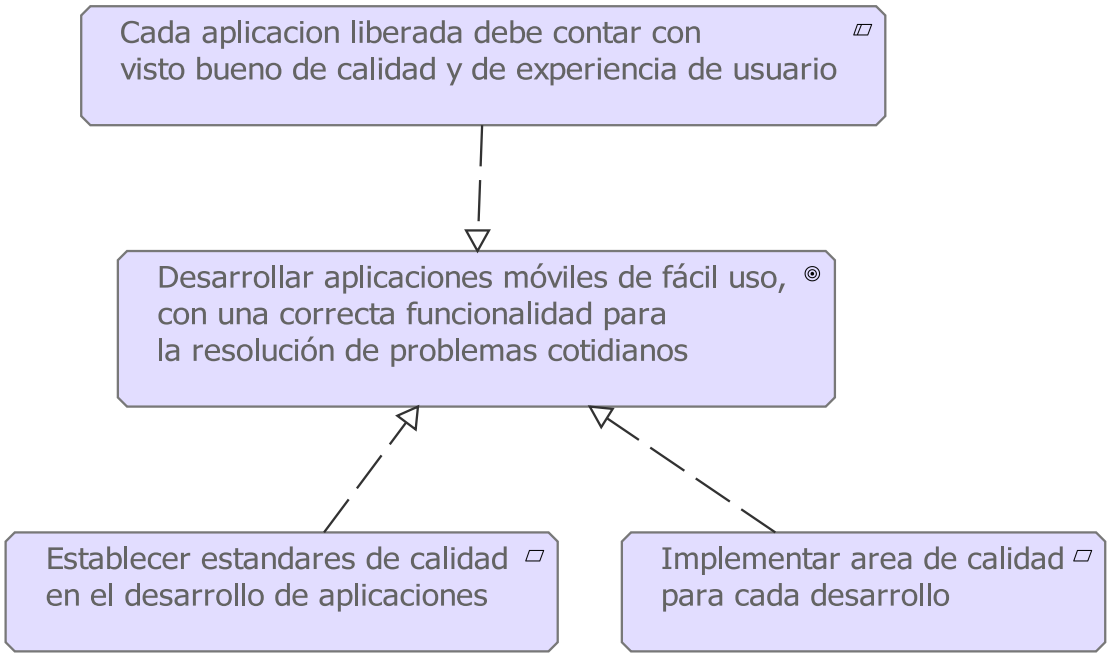


Figura 4-8.: Caso Realización de Objetivos

4.3.4. Punto de vista de contribución de objetivos

Este punto de vista permite el modelado de las relaciones de influencia entre objetivos y requisitos, mediante el resultado de estas vistas se puede analizar el impacto de las metas y detectar posibles conflictos en los objetivos y las partes interesadas. Los elementos para este punto de vista se observan en la figura **4-9**, el caso modelado propuesto se representa en la figura **4-10**

Modelo de contribución de objetivos

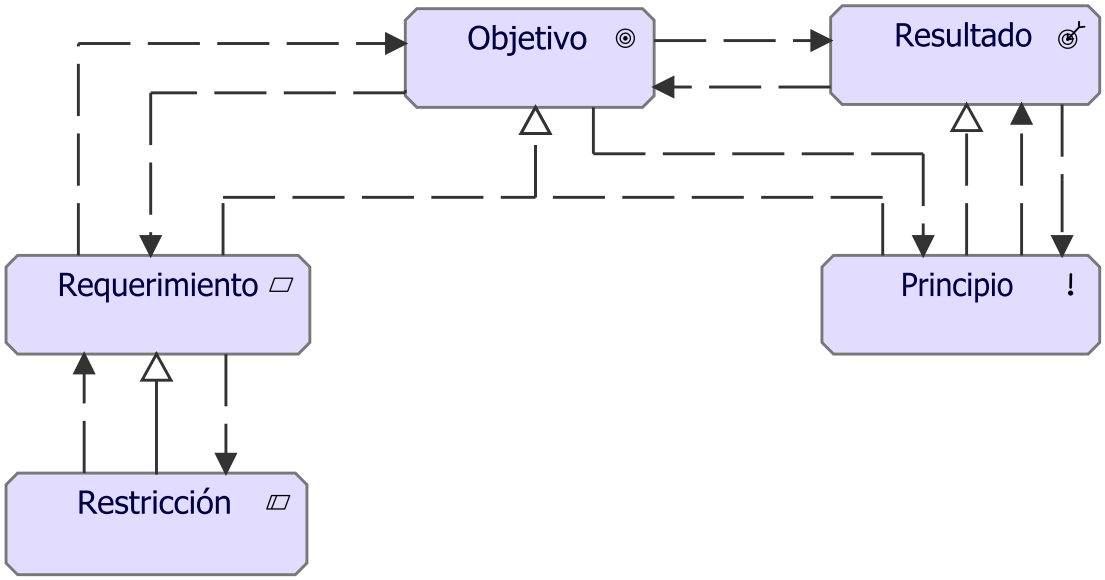


Figura 4-9.: Contribución de Objetivos

Caso de contribución de objetivos

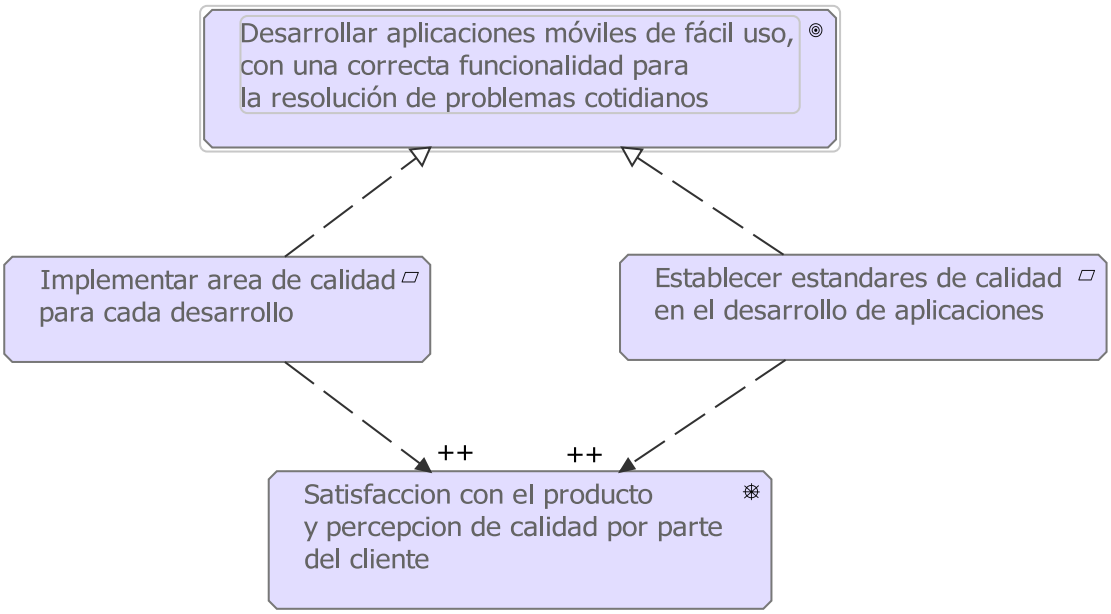


Figura 4-10.: Caso Contribución de Objetivos

4.3.5. Punto de vista de principios

Este punto de vista permite el modelado de los principios mas relevantes para la consecución de los objetivos, apoyándose a su ves en los requerimientos y restricciones del mismo. Los elementos para este punto de vista se observan en la figura **4-11**, el caso modelado propuesto se representa en la figura **4-12**

Modelo de principios

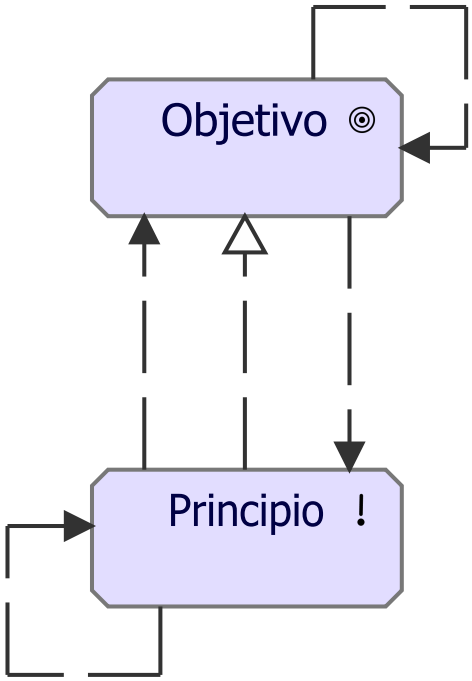


Figura 4-11.: Principios

Caso de principios

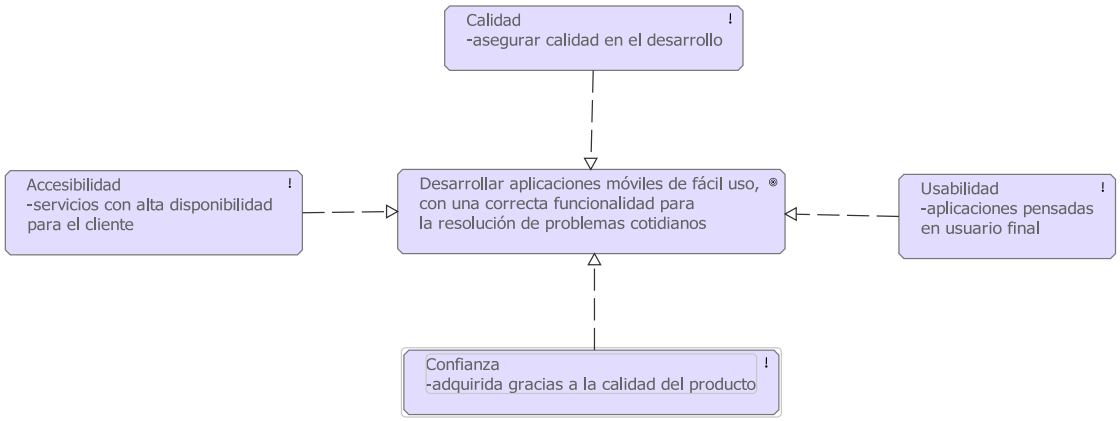


Figura 4-12.: Caso Principios

4.3.6. Punto de vista de realización de requerimientos

Mediante este punto de vista se permite modelar la realización de los requisitos por parte de actores y servicios empresariales, se tienen en cuenta elementos como los procesos y servicios necesarios para la realización de un objetivo. Los elementos para este punto de vista se observan en la figura **4-13**, el caso modelado propuesto se representa en la figura **4-14**

Modelo de realización de requerimientos

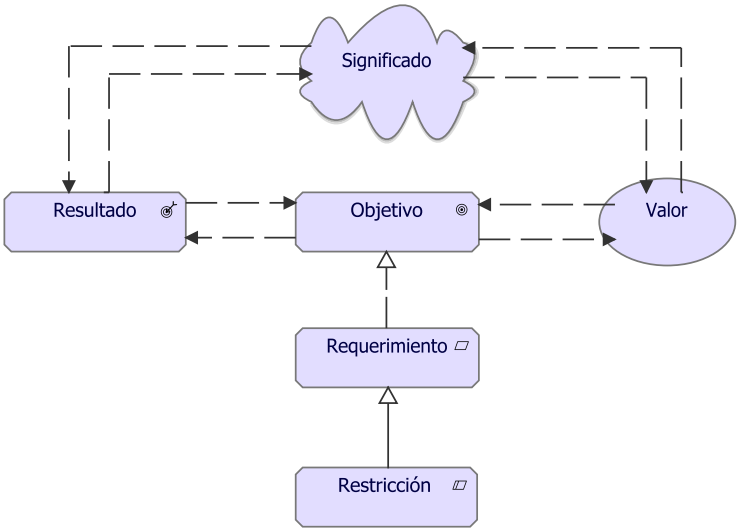


Figura 4-13.: Realización de Requerimientos

Caso de realización de requerimientos

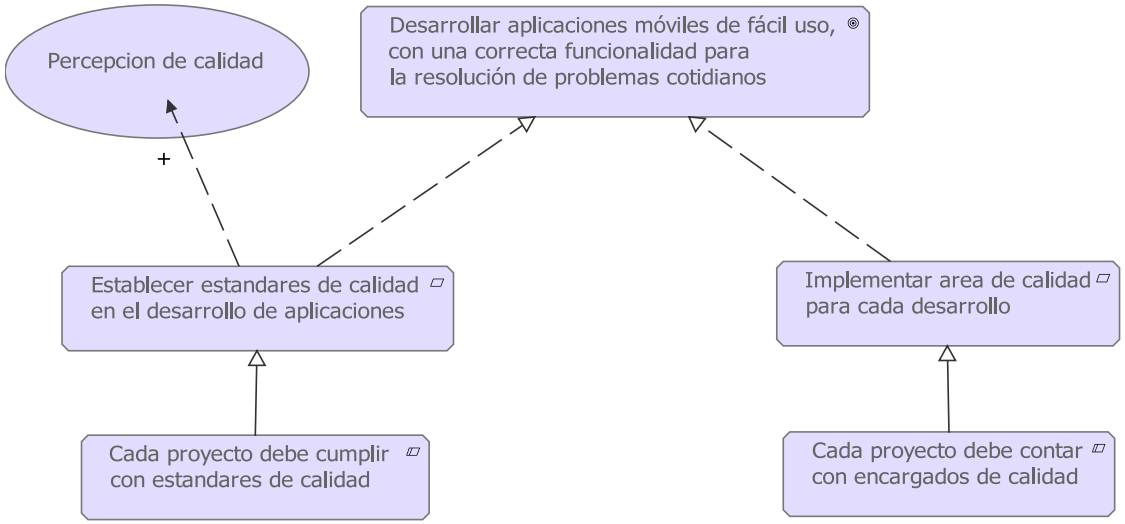


Figura 4-14.: Caso Realización de Requerimientos

4.3.7. Punto de vista de motivación

En este punto de vista se presenta un panorama completo del aspecto de la motivación con relación a los interesados, sus objetivos, requerimientos o restricciones, principios así como resultados con un significado y valor para la organización. Los elementos para este punto de vista se observan en la figura **4-15**, el caso modelado propuesto se representa en la figura **4-16**

Modelo de motivación

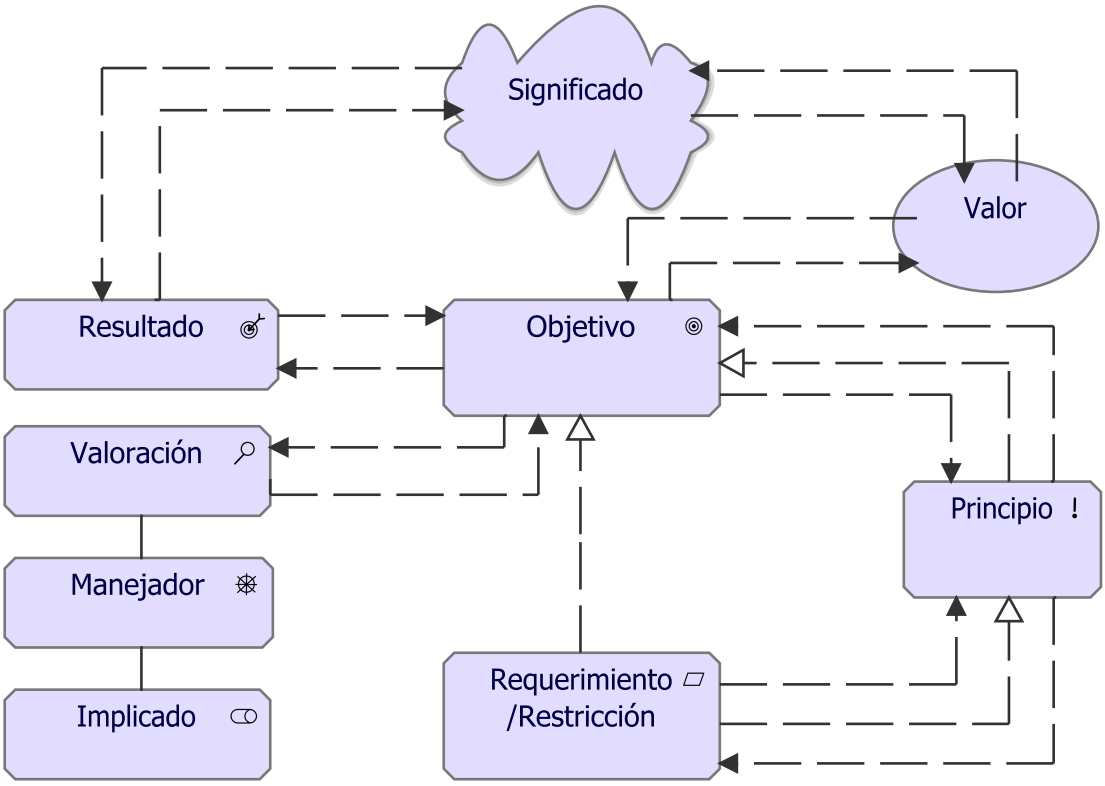


Figura 4-15.: Motivación

Caso de motivación

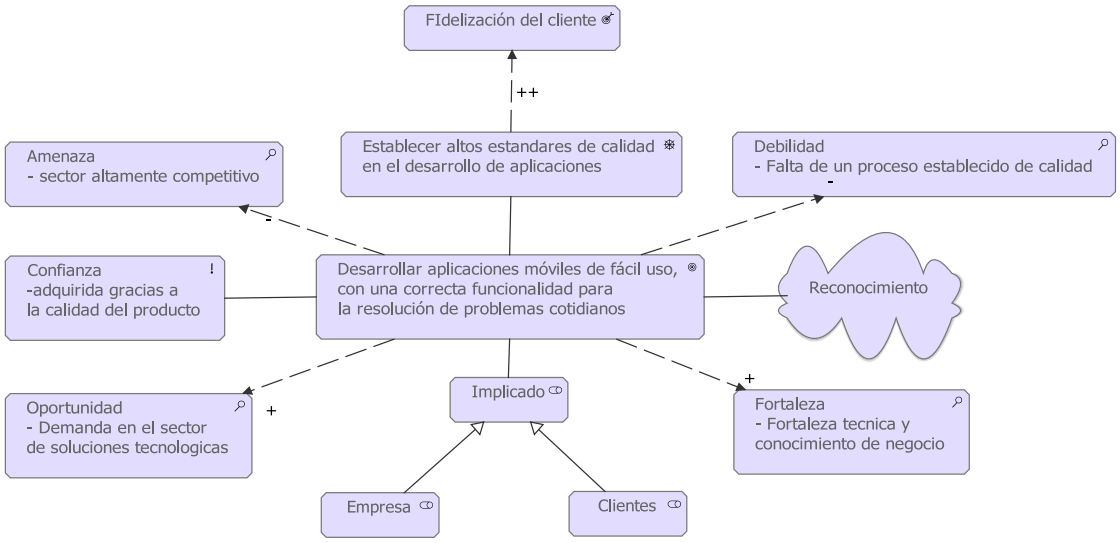


Figura 4-16.: Caso Motivación

4.4. Capa de estrategia

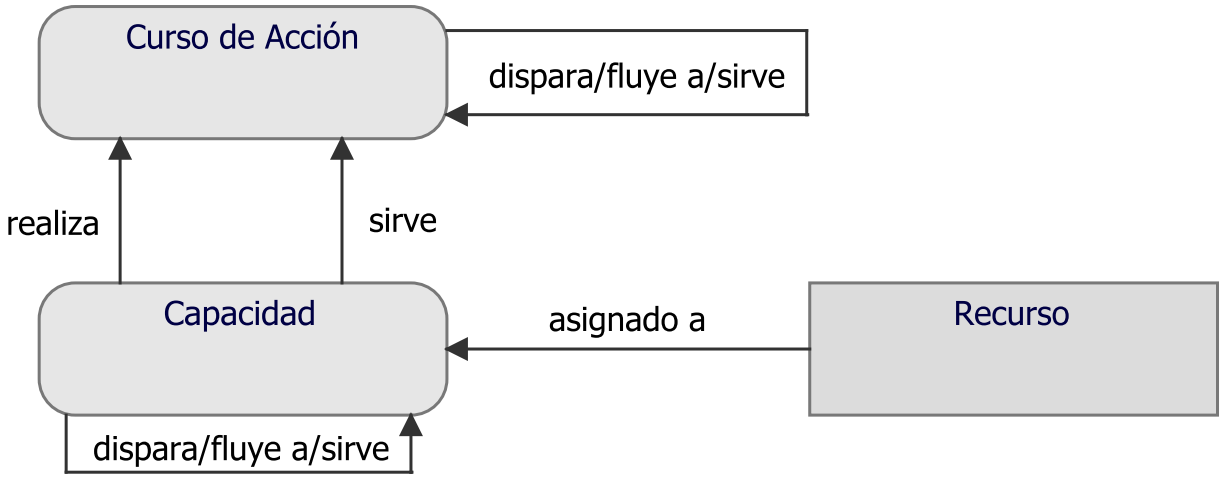


Figura 4-17.: Metamodelo de Estrategia

4.4.1. Introducción

Esta capa se encanta de aspectos estratégicos de la empresa, se presenta una perspectiva diferente sobre el modelado de la dirección estrategia de alto nivel y composición de la organización, se busca establecer diferentes planes estratégicos mediante procesos de negocio, usando conceptos propuestos en la capa de motivación para la identificaciones de los usos e interacciones con esta capa. [5]

4.4.2. Punto de vista de mapa de capacidad

En este punto de vista se crea una visión general de la estructura de las capacidades de la empresa mediante la identificación de los recursos y capacidades que llevan a un resultado concreto estratégico. Los elementos para este punto de vista se observan en la figura **4-18**, el caso modelado propuesto se representa en la figura **4-19**

Modelo de mapa de capacidad

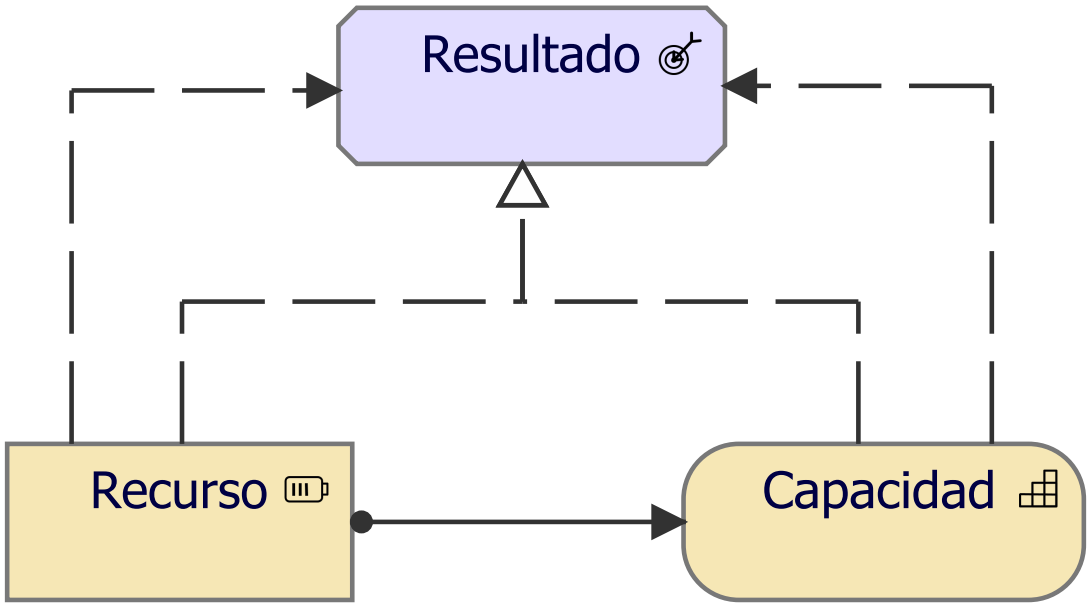


Figura 4-18.: Mapa de Capacidad

Caso de mapa de capacidad

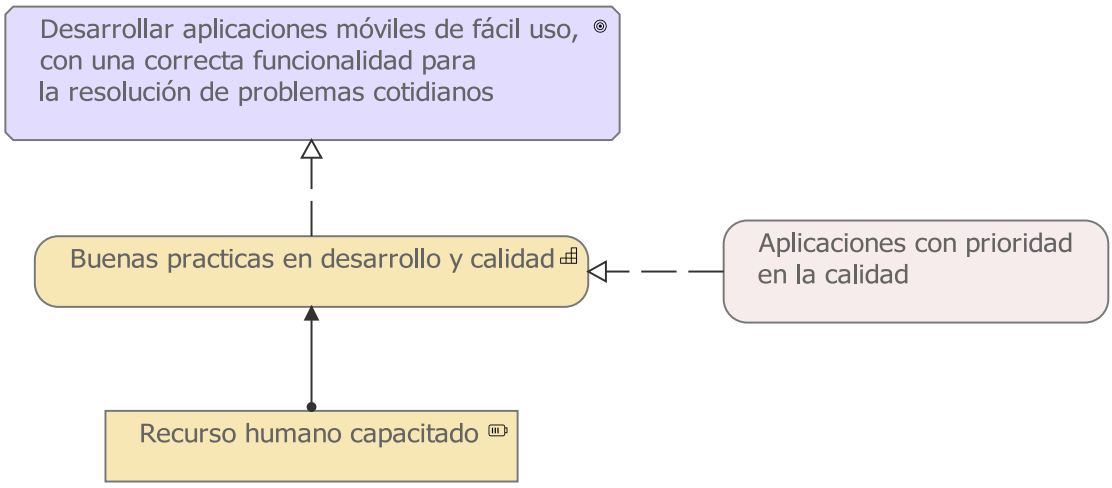


Figura 4-19.: Caso Mapa de Capacidad

4.4.3. Punto de vista de realización de resultado

En este punto de vista se vinculan conceptos de la capa motivacional, permitiendo dar un significado y valor a los elementos de recursos y capacidades en el planteamiento de un resultado de estrategia. Los elementos para este punto de vista se observan en la figura **4-20**, el caso modelado propuesto se representa en la figura **4-21**

Modelo de realización de resultado

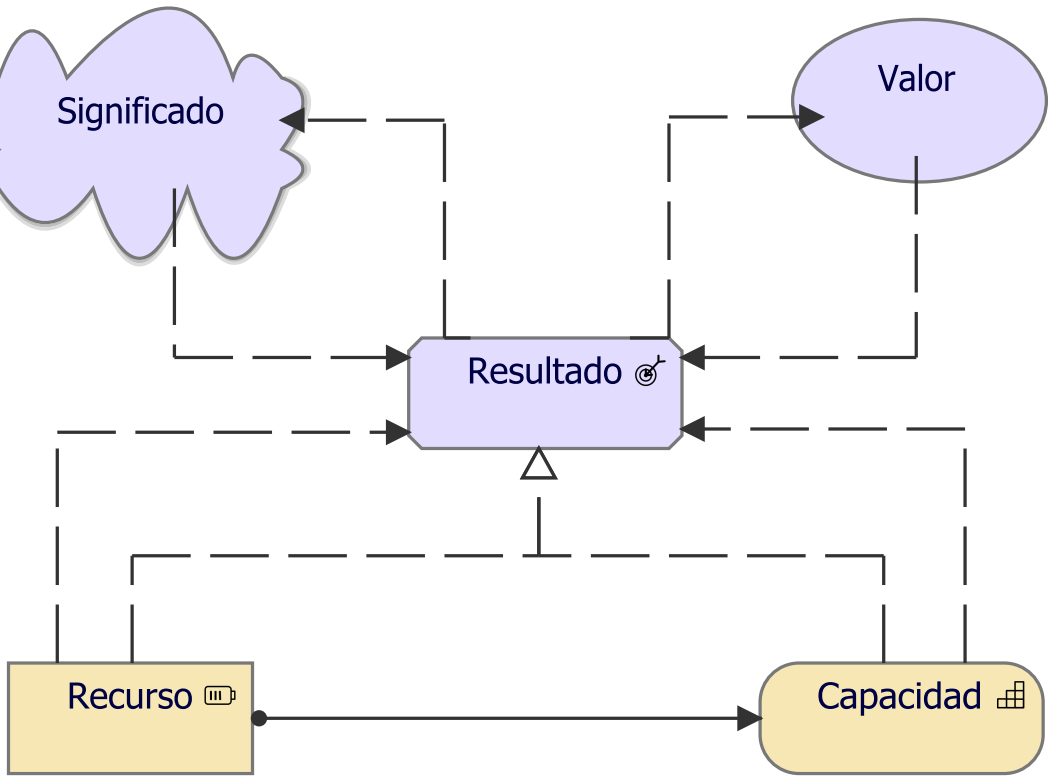


Figura 4-20.: Realización de Resultado

Caso de realización de resultado

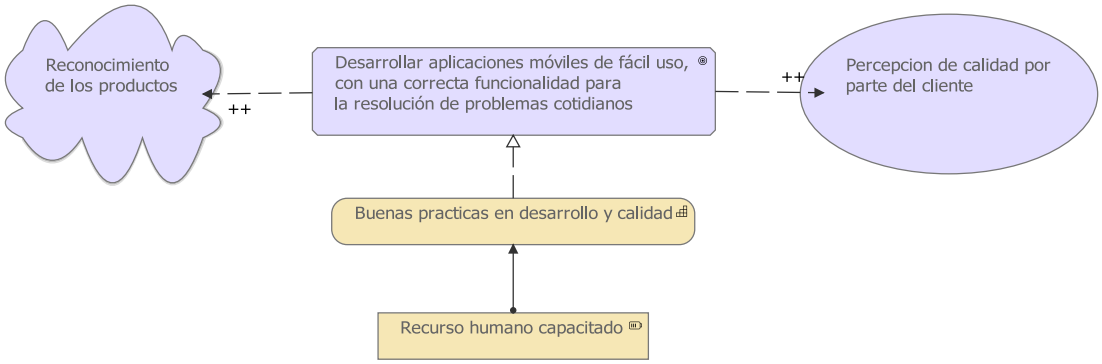


Figura 4-21.: Caso Realización de Resultado

4.4.4. Punto de vista de mapa de recurso

En este punto de vista los elementos de recurso y capacidad establecidos en las anteriores puntos, se vinculan a un paquete de trabajo, reuniendo un conjunto de actividades candidatas para el primer entregable de la arquitectura. Los elementos para este punto de vista se observan en la figura **4-22**, el caso modelado propuesto se representa en la figura **4-23**

Modelo de mapa de recurso

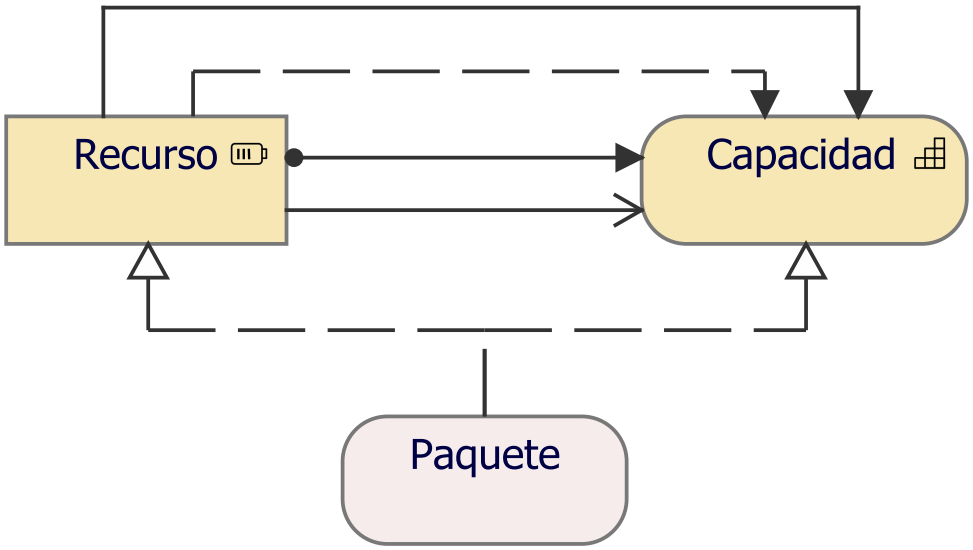


Figura 4-22.: Mapa de Recurso

Caso de mapa de recurso

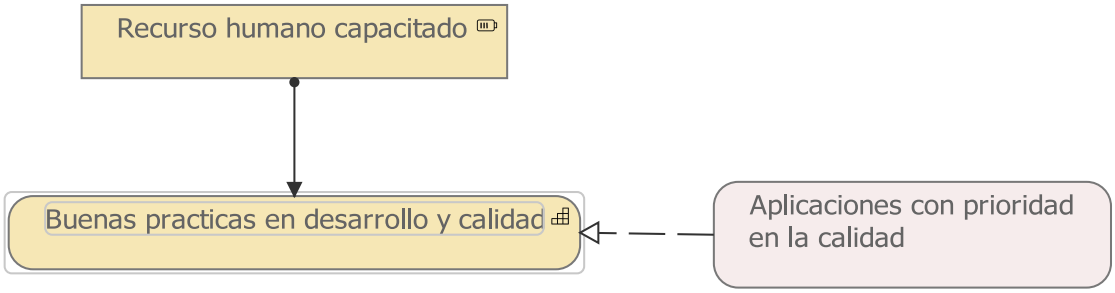


Figura 4-23.: Caso Mapa de Recurso

4.4.5. Punto de vista de flujo de valor

Este punto de vista incorpora a los implicados y un elemento que da valor a la estrategia incorporando una capacidad y un resultado empresariales, estos para llegar a un resultado estratégico. Los elementos para este punto de vista se observan en la figura **4-24**, el caso modelado propuesto se representa en la figura **4-25**

Modelo de flujo de valor

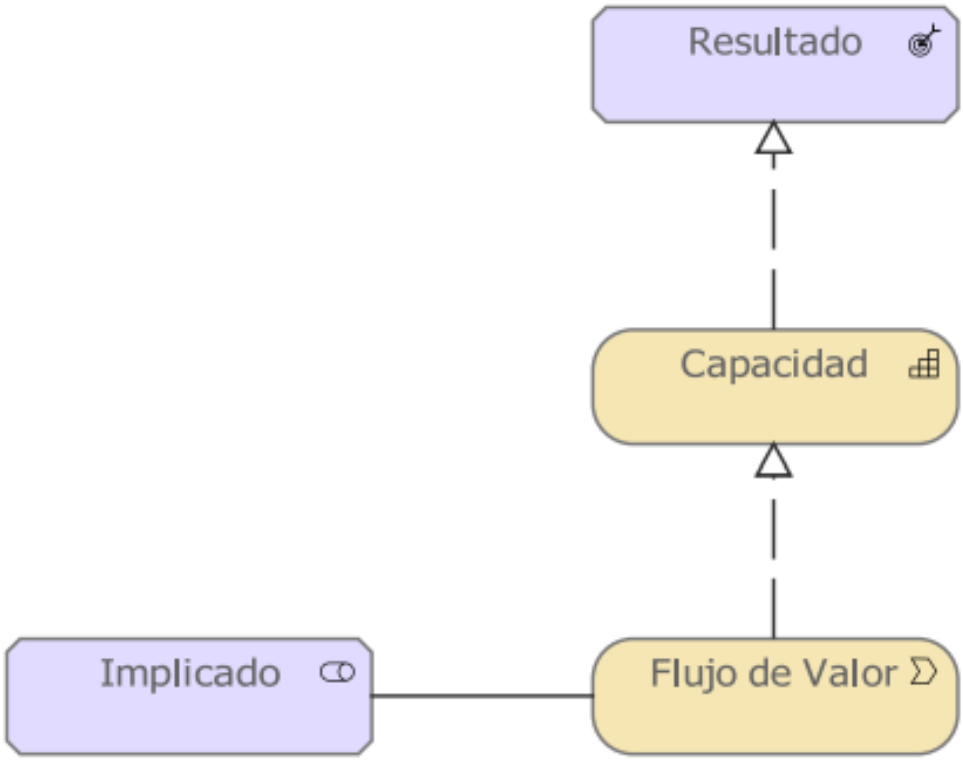


Figura 4-24.: Flujo de Valor

Caso de flujo de valor

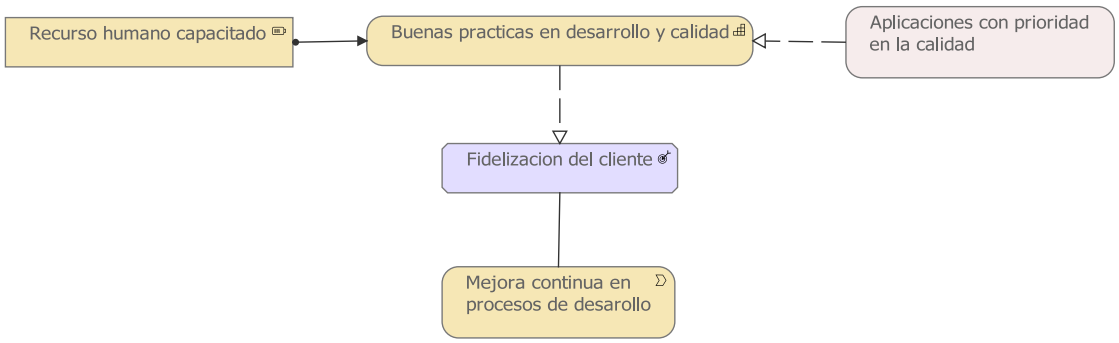


Figura 4-25.: Caso Flujo de Valor

4.4.6. Punto de vista de estrategia

En este punto de vista se organiza una visión general de la estrategia a alto nivel mediante el establecimiento de un curso de acción que involucra los recursos y capacidades para llegar a un resultado establecido previamente. Los elementos para este punto de vista se observan en la figura **4-26**, el caso modelado propuesto se representa en la figura **4-27**

Modelo de estrategia

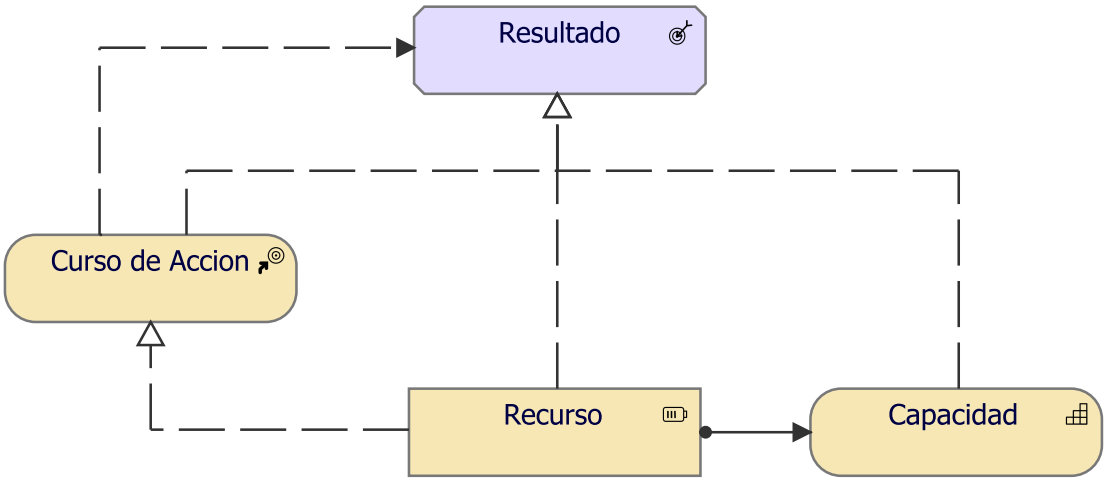


Figura 4-26.: Estrategia

Caso de estrategia

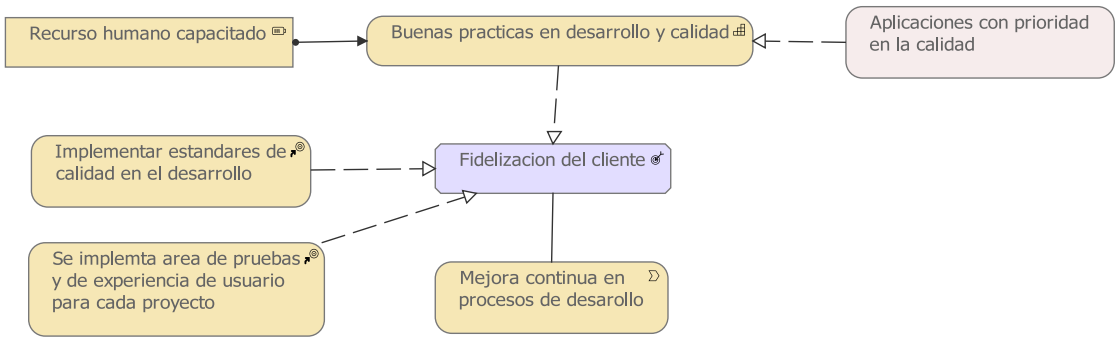


Figura 4-27.: Caso Estrategia

4.5. Capa de negocio

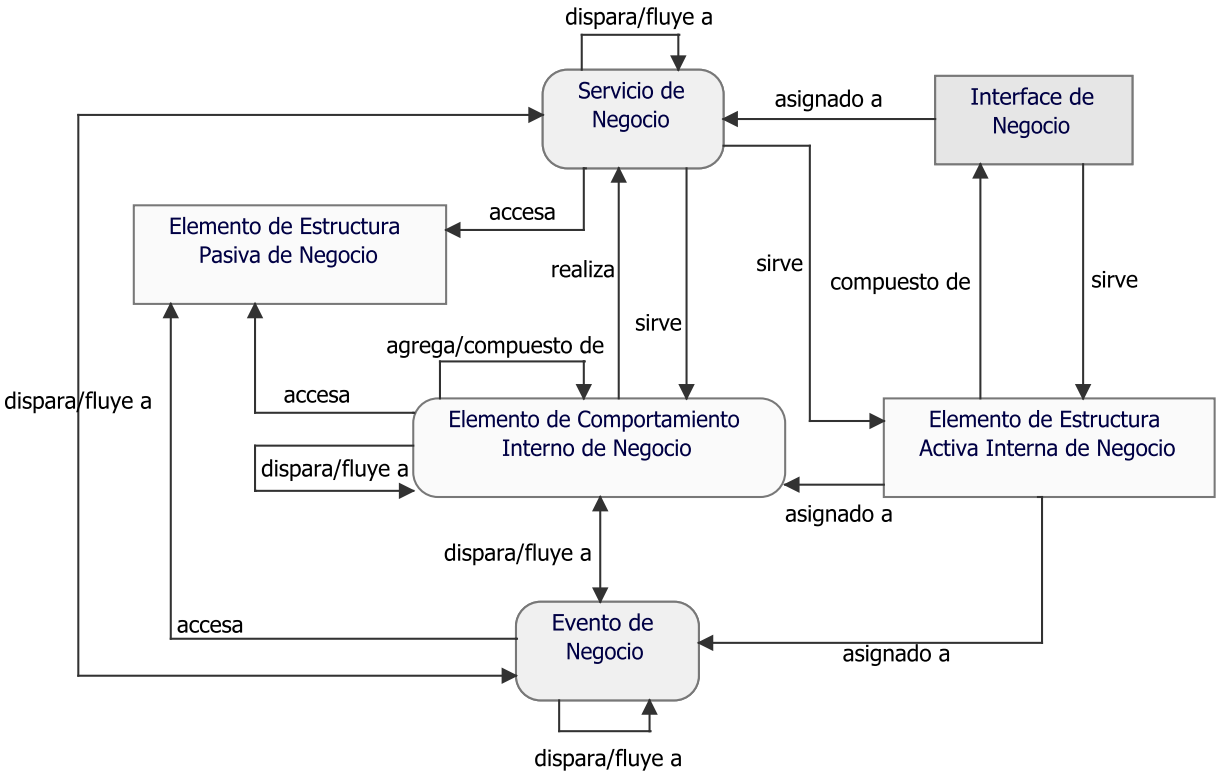


Figura 4-28.: Metamodelo de Estrategia

4.5.1. Introducción

En esta capa se presentan los conceptos que definen la estructura estática de la organización, implicando componentes e interacciones como actores, roles, funciones de negocio, entre otros. Se plantea un diseño orientado al servicio, funciones de procesos de negocio internos y externos de la organización. Dentro de los puntos de vista de esta capa se presentan las relaciones internas y externas, estableciendo procesos de comunicación entre estas mediante interfaces que permiten el accesos a servicios mediante colaboraciones dadas por las entidades de la organización. [5]

4.5.2. Punto de vista de organización

En este punto de vista se modela la estructura interna de la organización, permitiendo identificar las responsabilidades, mediante actores, roles y colaboraciones entre estos. Los elementos para este punto de vista se observan en la figura **4-29**, el caso modelado propuesto se representa en la figura **4-30**

Modelo de organización

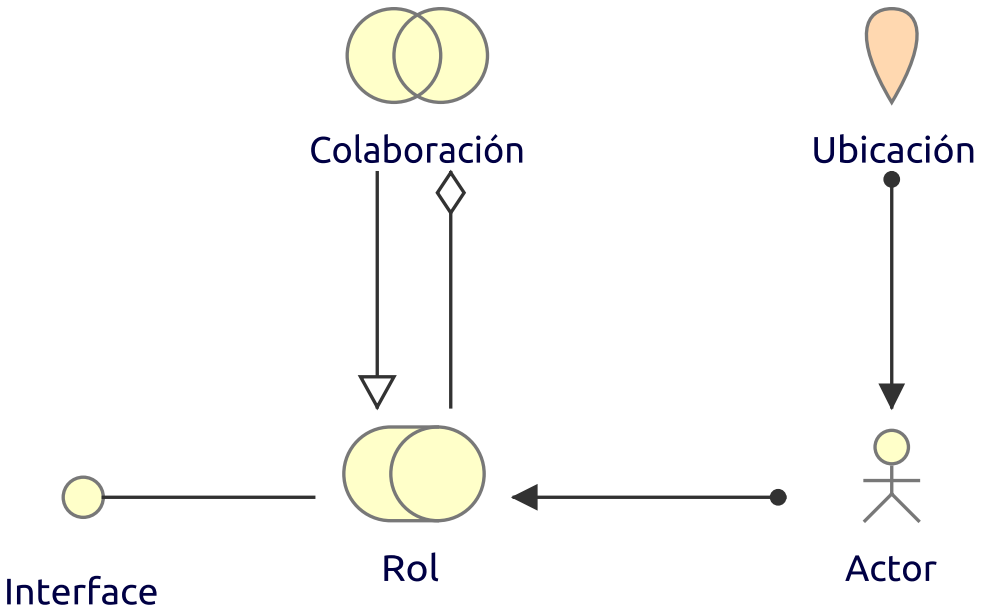


Figura 4-29.: Organización

Caso de organización

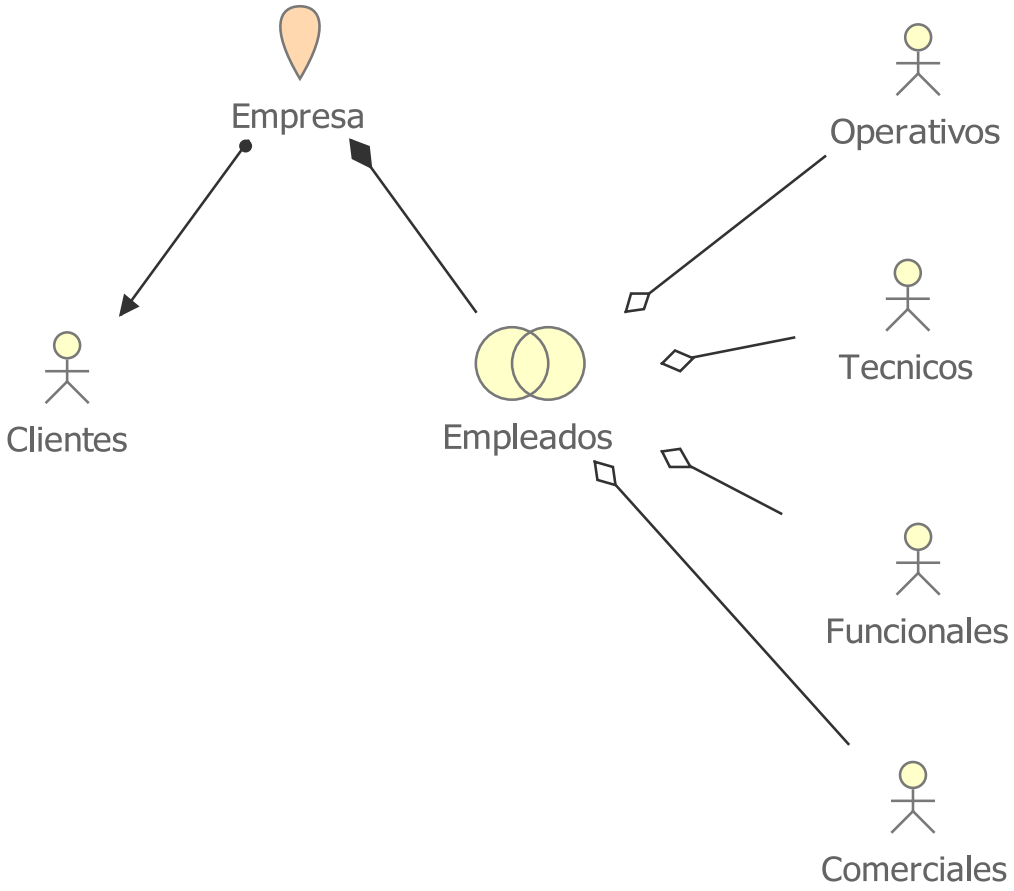


Figura 4-30.: Caso Organización

4.5.3. Punto de vista de cooperación de actor

Este punto de vista se centra en las relaciones de los actores entre si y con su entorno, estableciendo posibles dependencias externas y colaboraciones entre estos. Los elementos para este punto de vista se observan en la figura **4-31**, el caso modelado propuesto se representa en la figura **4-32**

Modelo de cooperación de actor

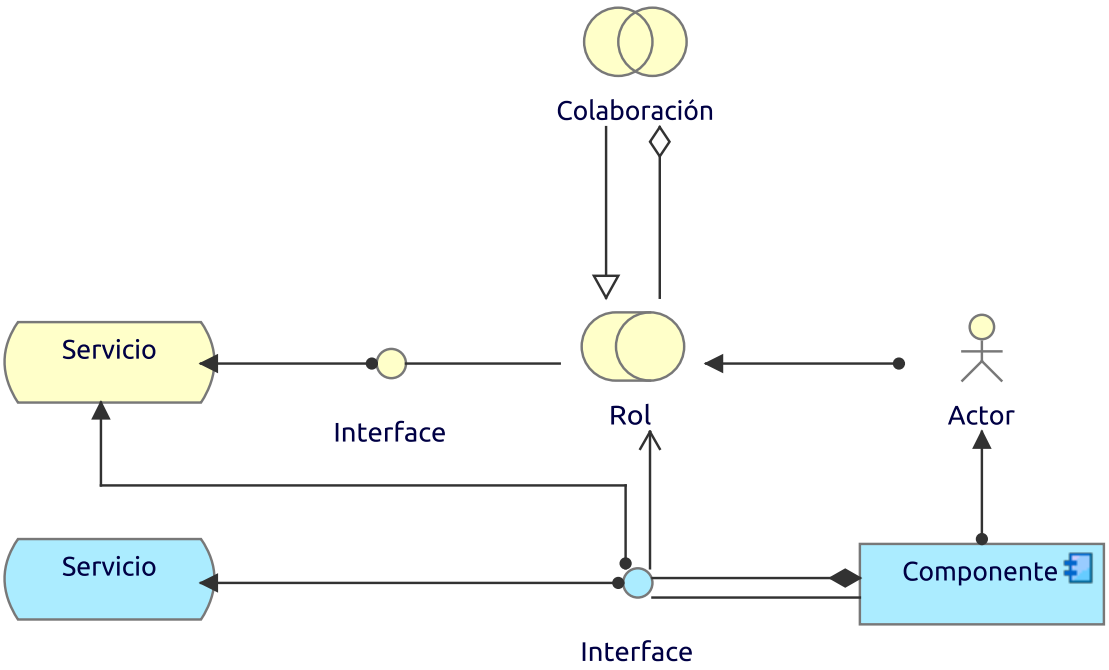


Figura 4-31.: Cooperación de Actor

Caso de cooperación de actor

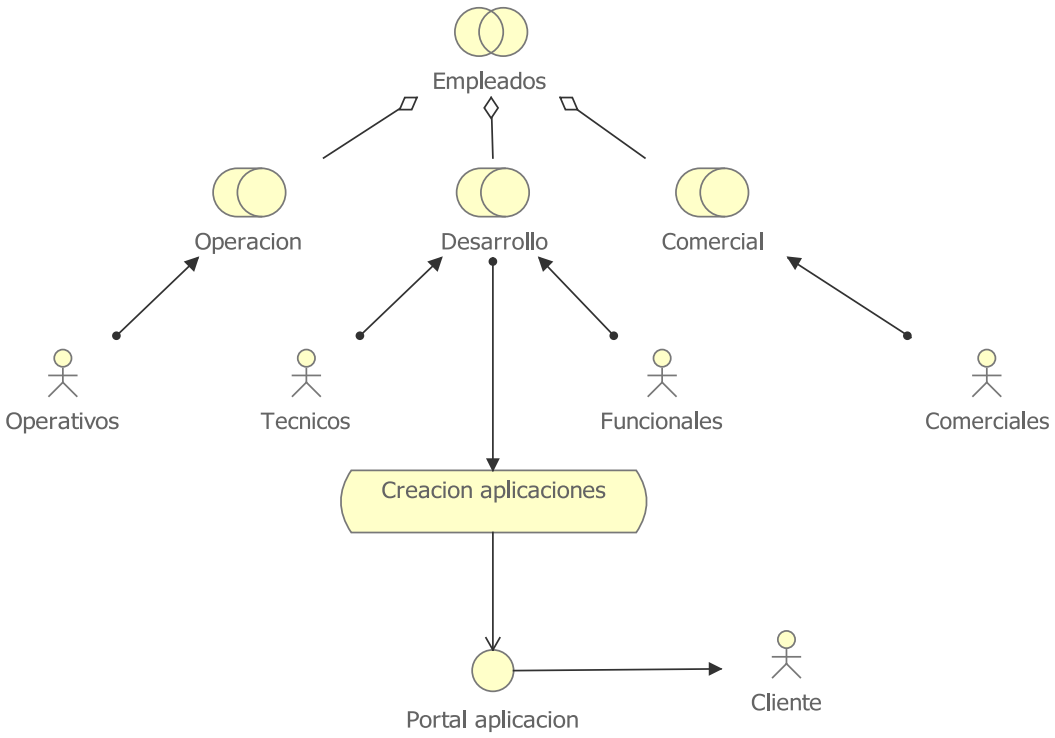


Figura 4-32.: Caso Cooperación de Actor

4.5.4. Punto de vista de función de negocio

En este punto de vista se establece una función del negocio de la organización, así como las relaciones existentes entre estos flujos y el rol que ejecuta la función, esto permite ofrecer una visión de alto nivel de las operaciones de la organización. Los elementos para este punto de vista se observan en la figura **4-33**, el caso modelado propuesto se representa en la figura **4-34**

Modelo de función de negocio

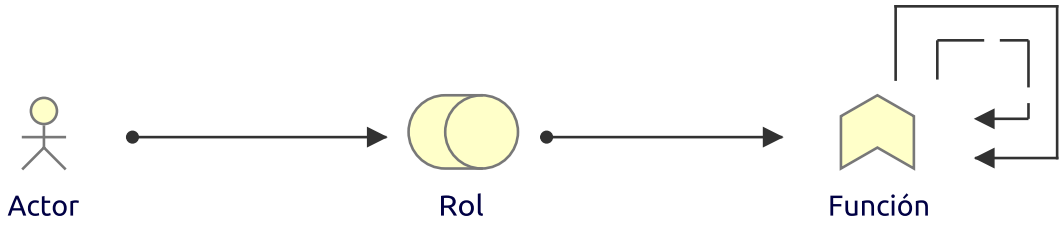


Figura 4-33.: Función de Negocio

Caso de función de negocio

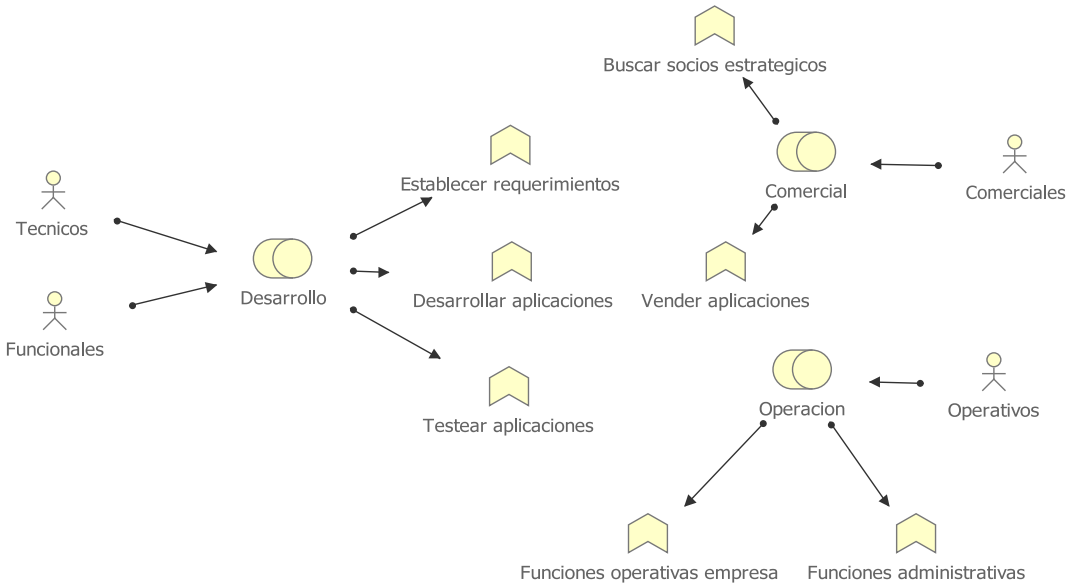


Figura 4-34.: Caso Función de Negocio

4.5.5. Punto de vista de proceso de negocio

Este punto de vista muestra la estructura y composición de alto nivel de los procesos de la organización, contiene elementos relacionados como eventos asociados a los procesos o funciones, asi como objetos y representación de los servicios, también incorpora actores y roles para establecer responsabilidades de. Los elementos para este punto de vista se observan en la figura **4-35**, el caso modelado propuesto se representa en la figura **4-36**

Modelo de proceso de negocio

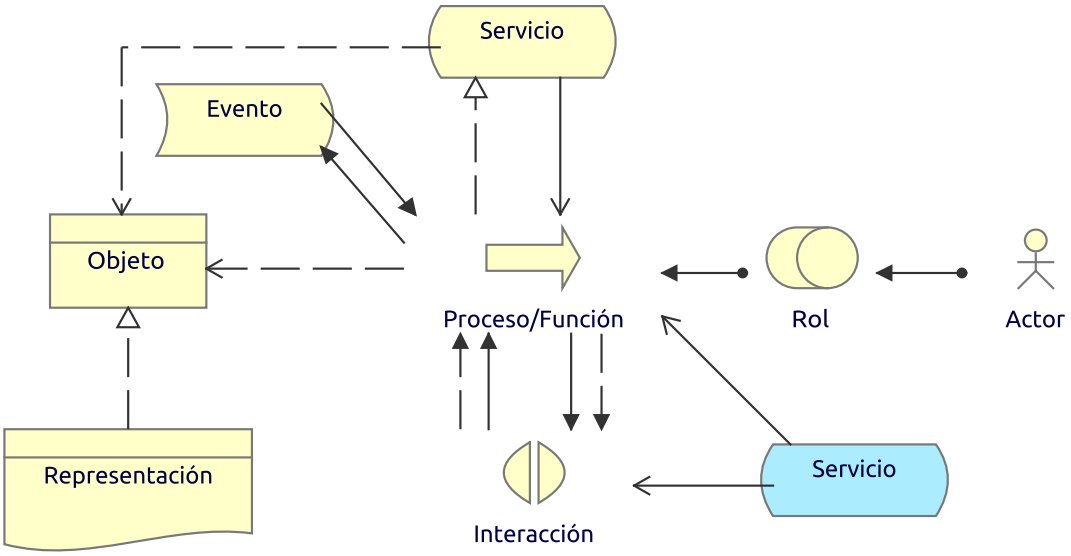


Figura 4-35.: Proceso de Negocio

Caso de proceso de negocio

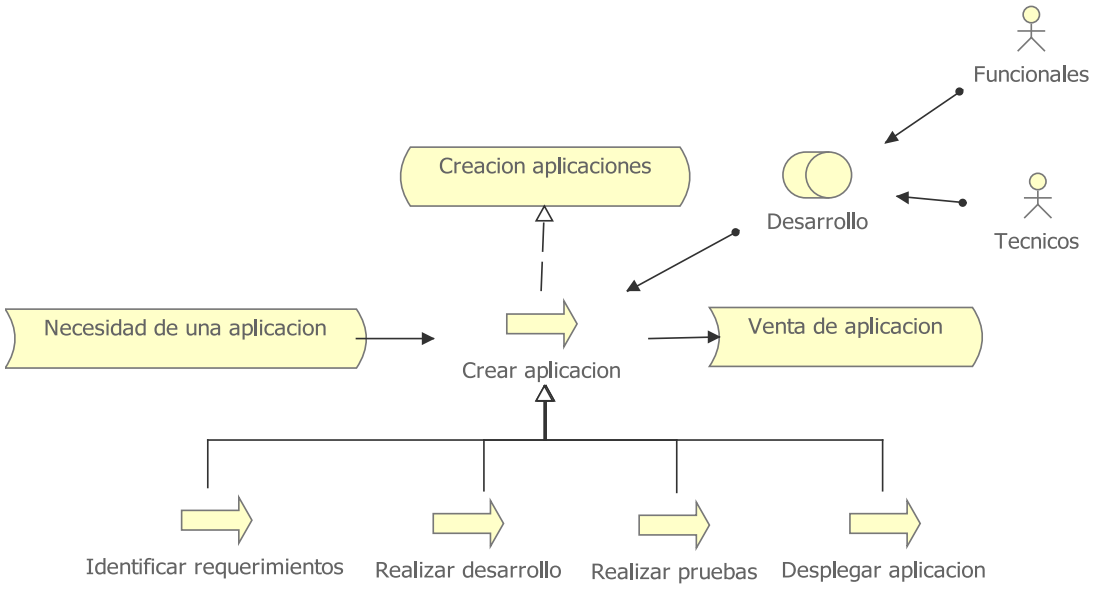


Figura 4-36.: Caso Proceso de Negocio

4.5.6. Punto de vista de cooperación de proceso de negocio

Este punto de vista se utiliza para mostrar relaciones de uno o mas procesos organizacionales entre si o con su entorno, agregando el elemento de localización que relaciona actores y objetos de negocio. Los elementos para este punto de vista se observan en la figura **4-37**, el caso modelado propuesto se representa en la figura **4-38**

Modelo de cooperación de proceso de negocio

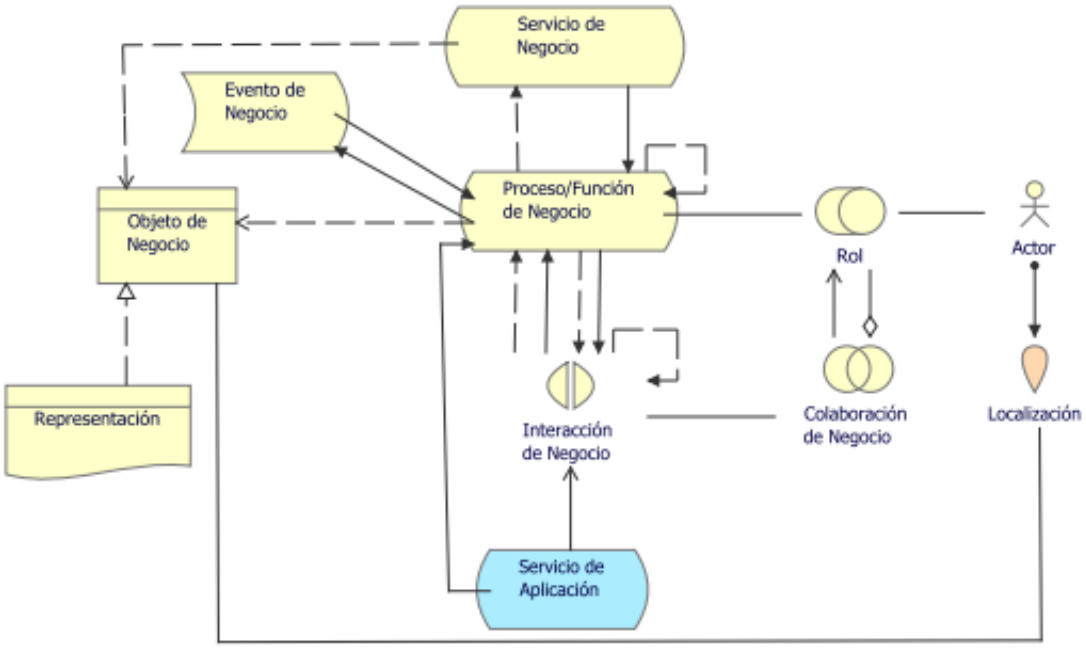


Figura 4-37.: Cooperación de Proceso de Negocio

Caso de cooperación de proceso de negocio

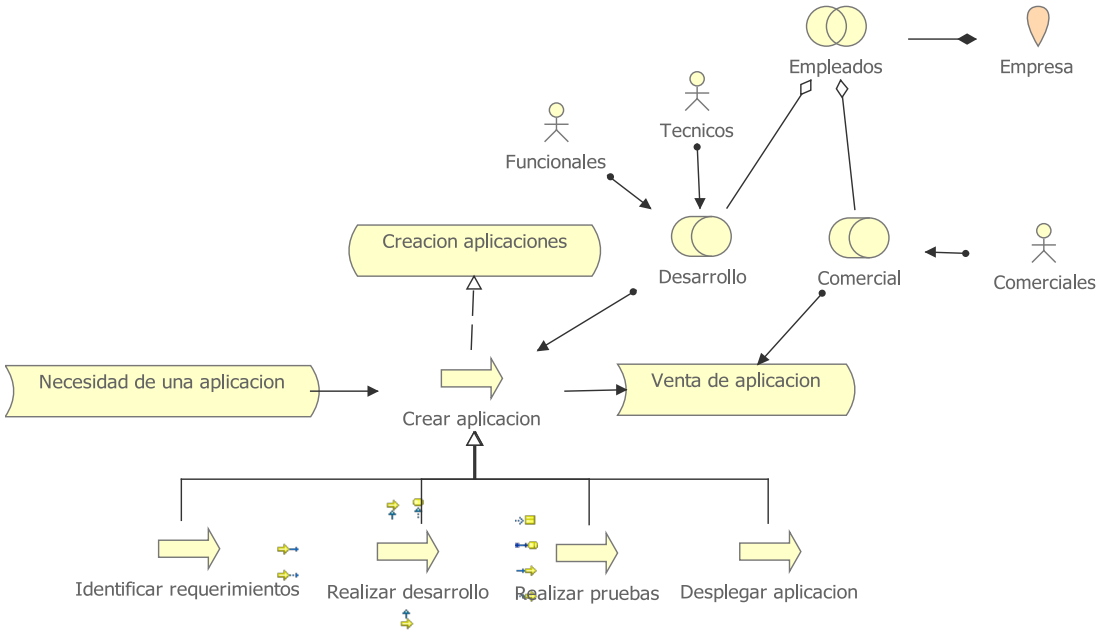


Figura 4-38.: Caso Cooperación de Proceso de Negocio

4.5.7. Punto de vista de producto

Este punto de vista incorpora las posibles colaboraciones entre los diferentes roles asociados a un proceso o función de la organización, también incorpora elementos enfocados a la capa de aplicación. Los elementos para este punto de vista se observan en la figura **4-39**, el caso modelado propuesto se representa en la figura **4-40**

Modelo de producto

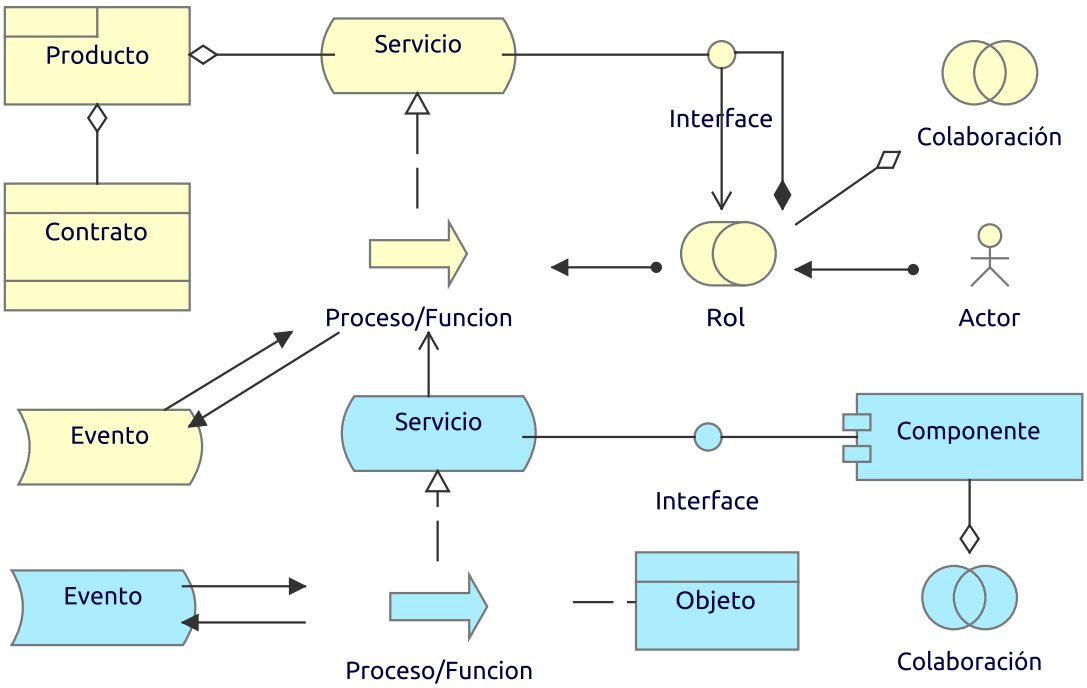


Figura 4-39.: Producto

Caso de producto

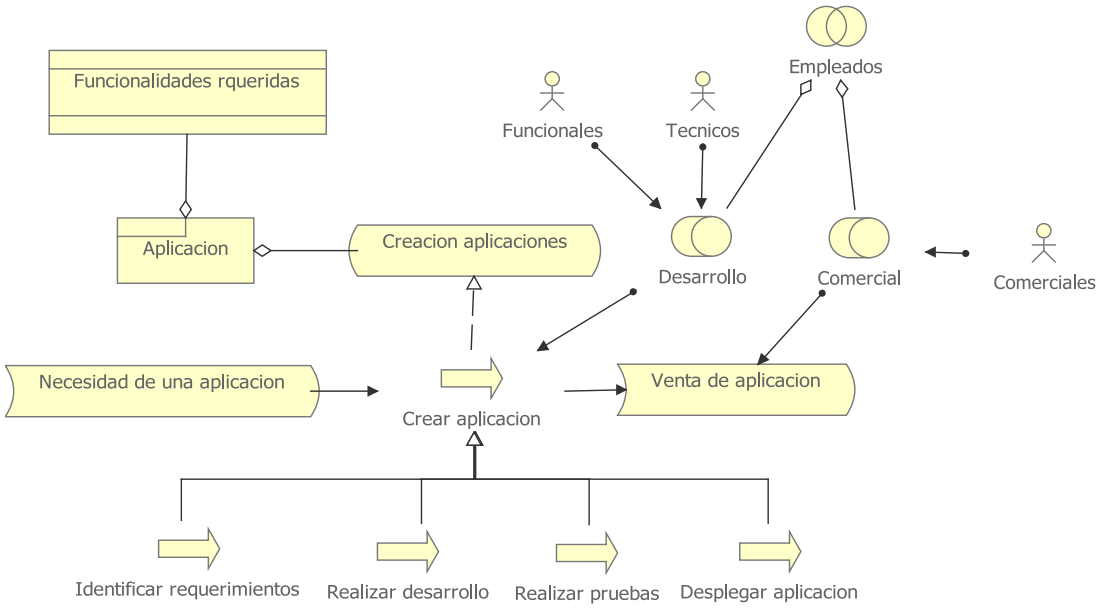


Figura 4-40.: Caso Producto

4.6. Capa de aplicación

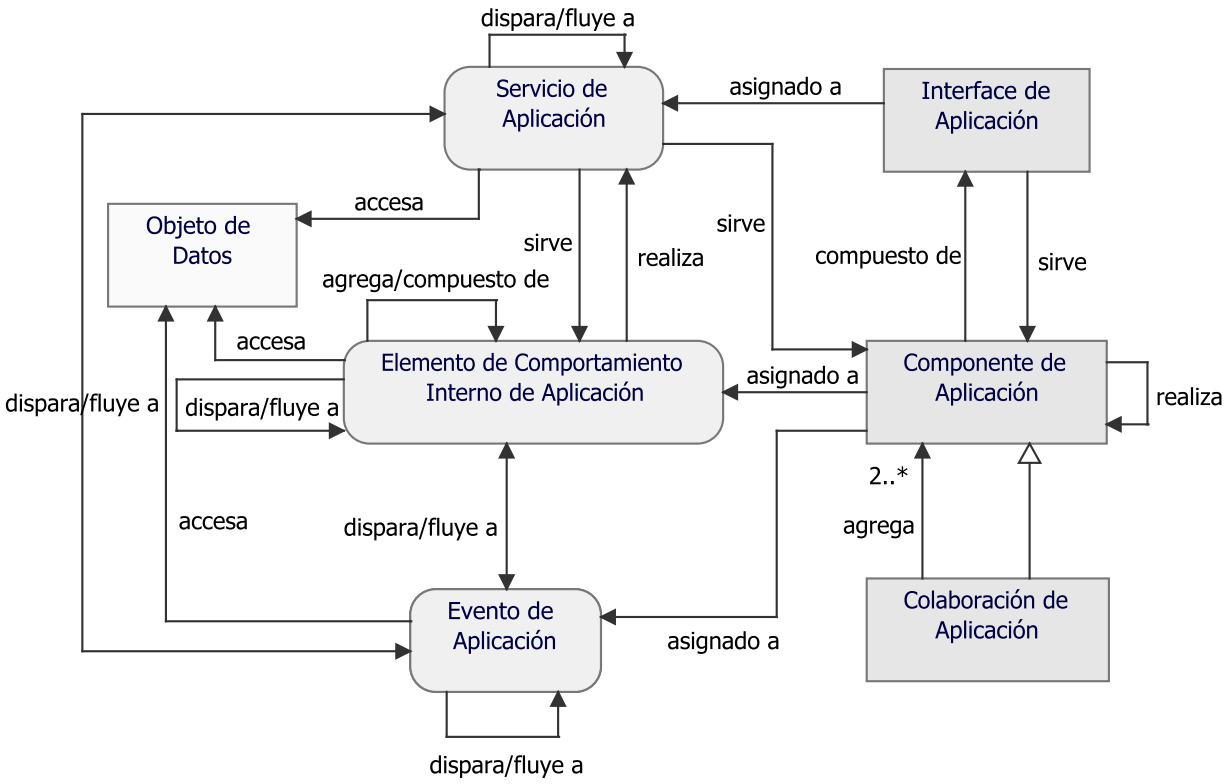


Figura 4-41.: Metamodelo de Estrategia

4.6.1. Introducción

En esta capa se abordan entidades como componentes y objeto de datos, así como las relaciones y comportamientos a nivel interno y externo con la organización, mediante servicios y funciones de aplicación. La interrelación entre componentes es fundamental en la arquitectura de la aplicación, teniendo en cuenta el concepto de colaboración para este fin y el concepto de interfaz como el canal logico mediante el cual se accede a un servicio del componente. [5]

4.6.2. Punto de vista de comportamiento de aplicación

En este punto de vista describe el comportamiento interno de la aplicación, apoyándose en elementos como el componente, el servicio la función y la colaboración entre los diferentes elementos. Los elementos para este punto de vista se observan en la figura **4-42**, el caso modelado propuesto se representa en la figura **4-43**

Modelo de comportamiento de aplicación

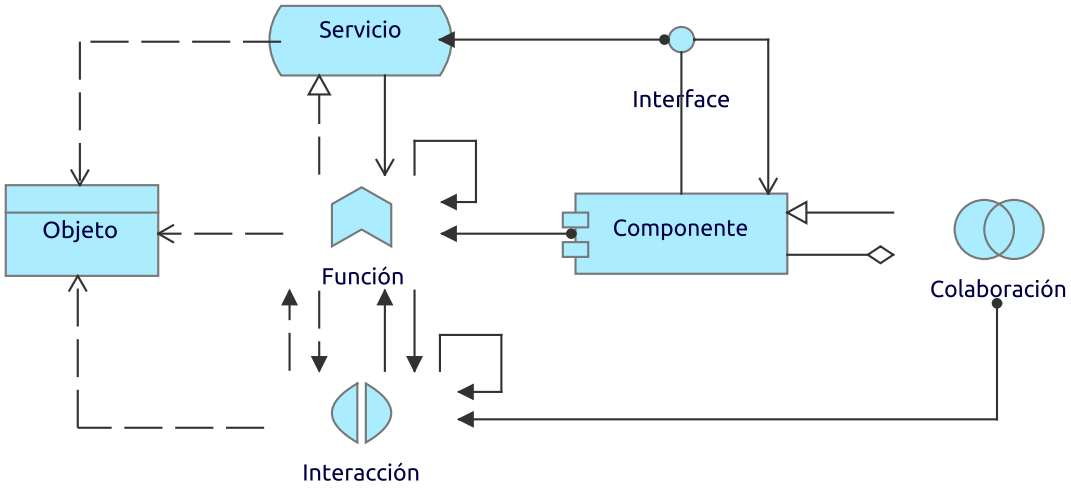


Figura 4-42.: Comportamiento de Aplicación

Caso de comportamiento de aplicación

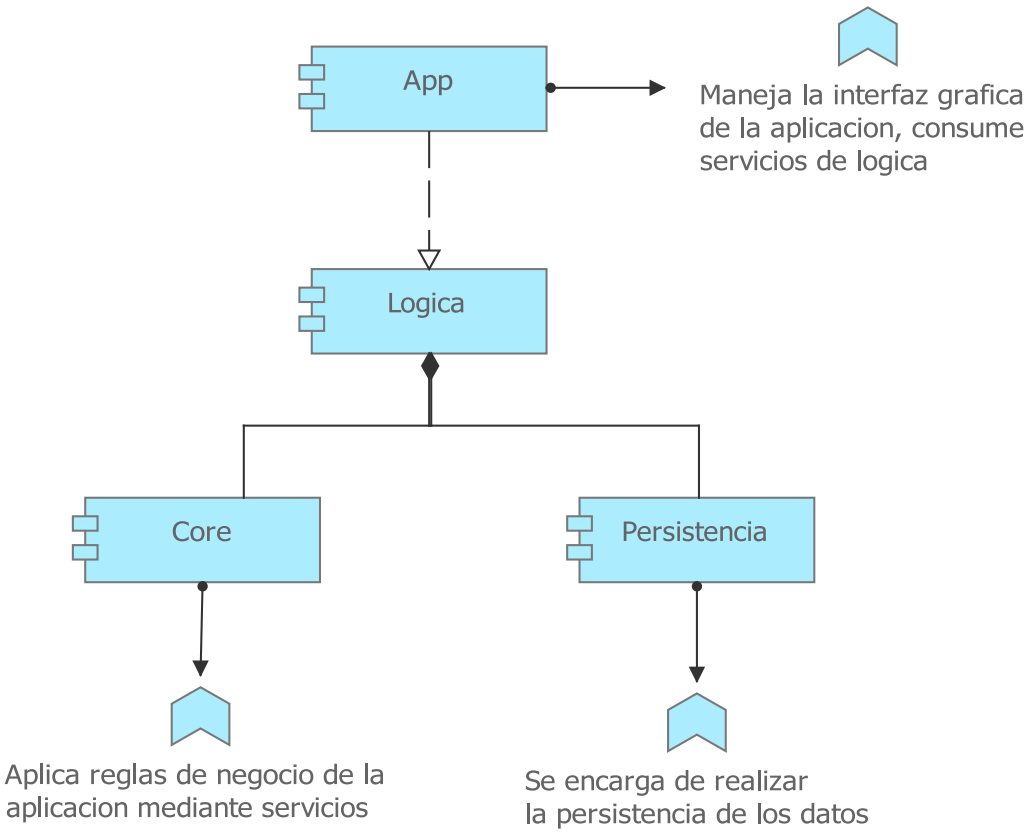


Figura 4-43.: Caso Comportamiento de Aplicación

4.6.3. Punto de vista de cooperación de aplicación

En este punto de vista se crea un entorno general de la aplicación, se busca la cooperación interna dentro de la organización mediante la orquestación de servicios que apoyan al proceso del negocio. Los elementos para este punto de vista se observan en la figura **4-44**, el caso modelado propuesto se representa en la figura **4-45**

Modelo de cooperación de aplicación

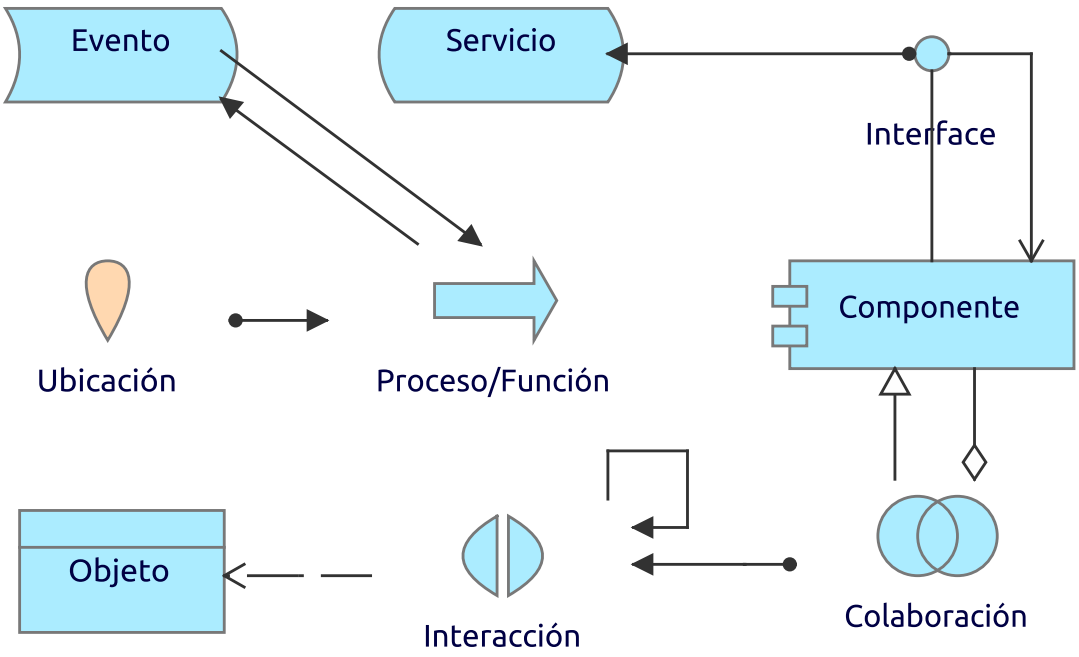


Figura 4-44.: Cooperación de Aplicación

Caso de cooperación de aplicación

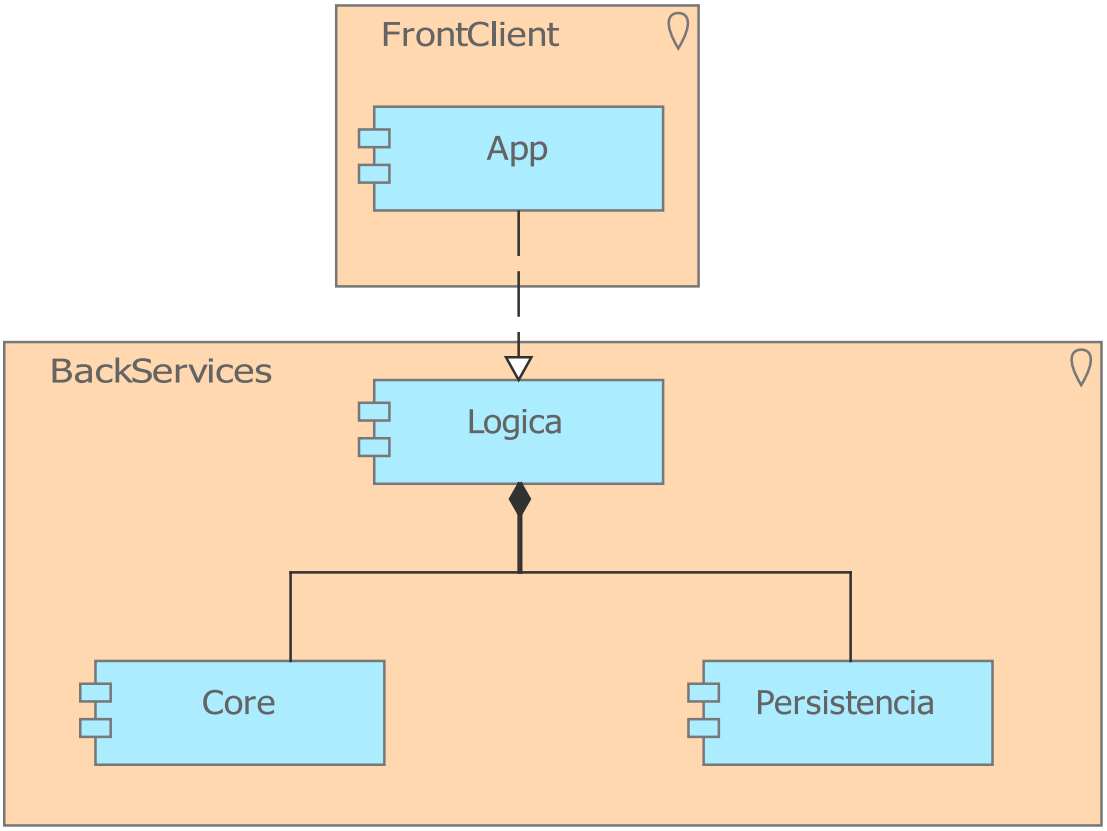


Figura 4-45.: Caso Cooperación de Aplicación

4.6.4. Punto de vista de estructura de aplicación

Este punto de vista se utiliza para realizar un diseño de la estructura principal de los componentes que forma la aplicación, estableciendo relaciones entre estos mediante interfaces. Los elementos para este punto de vista se observan en la figura **4-46**, el caso modelado propuesto se representa en la figura **4-47**

Modelo de estructura de aplicación

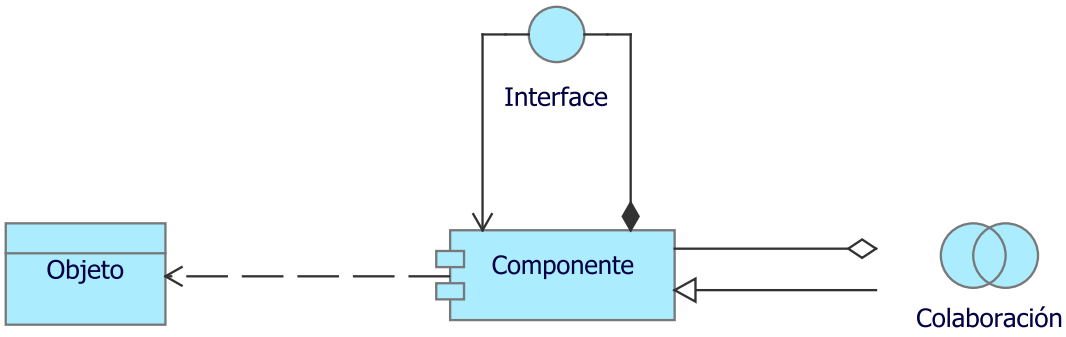


Figura 4-46.: Estructura de Aplicación

Caso de estructura de aplicación

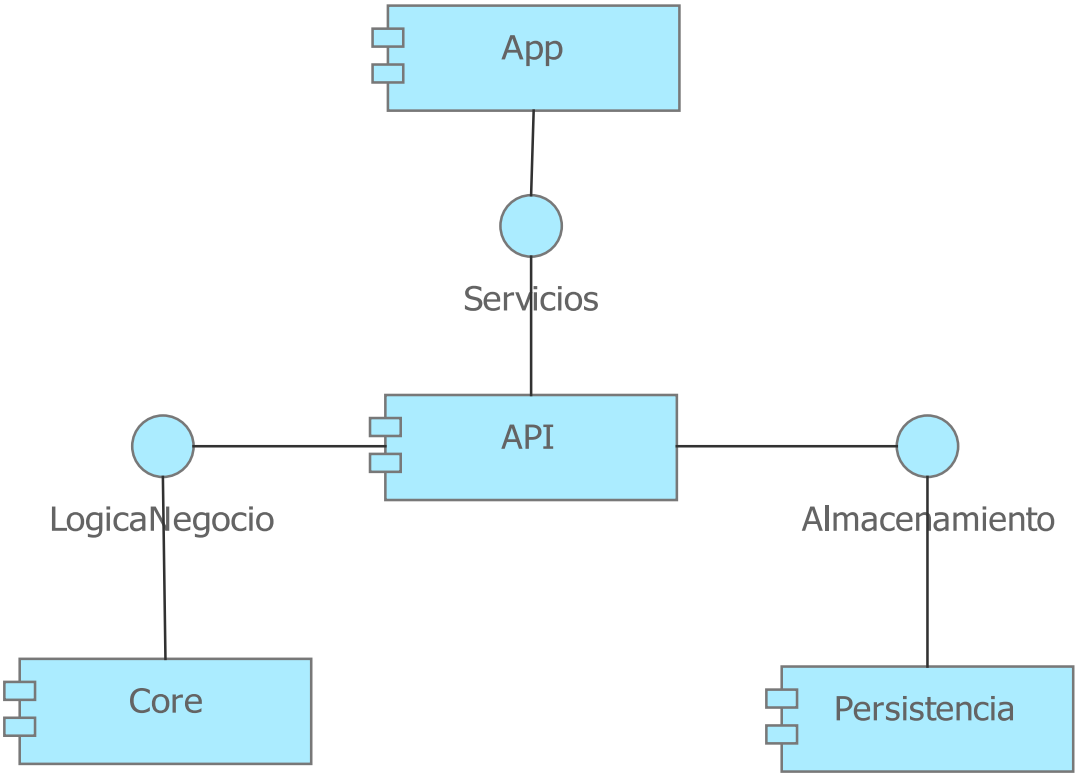


Figura 4-47.: Caso Estructura de Aplicación

4.6.5. Punto de vista de uso de aplicación

Este punto de vista permiten asociar los componentes y servicios de aplicación con elementos de la capa de negocio como eventos y procesos empresariales y muestra como son soportados por estos elementos de aplicación. Los elementos para este punto de vista se observan en la figura **4-48**, el caso modelado propuesto se representa en la figura **4-49**

Modelo de uso de aplicación

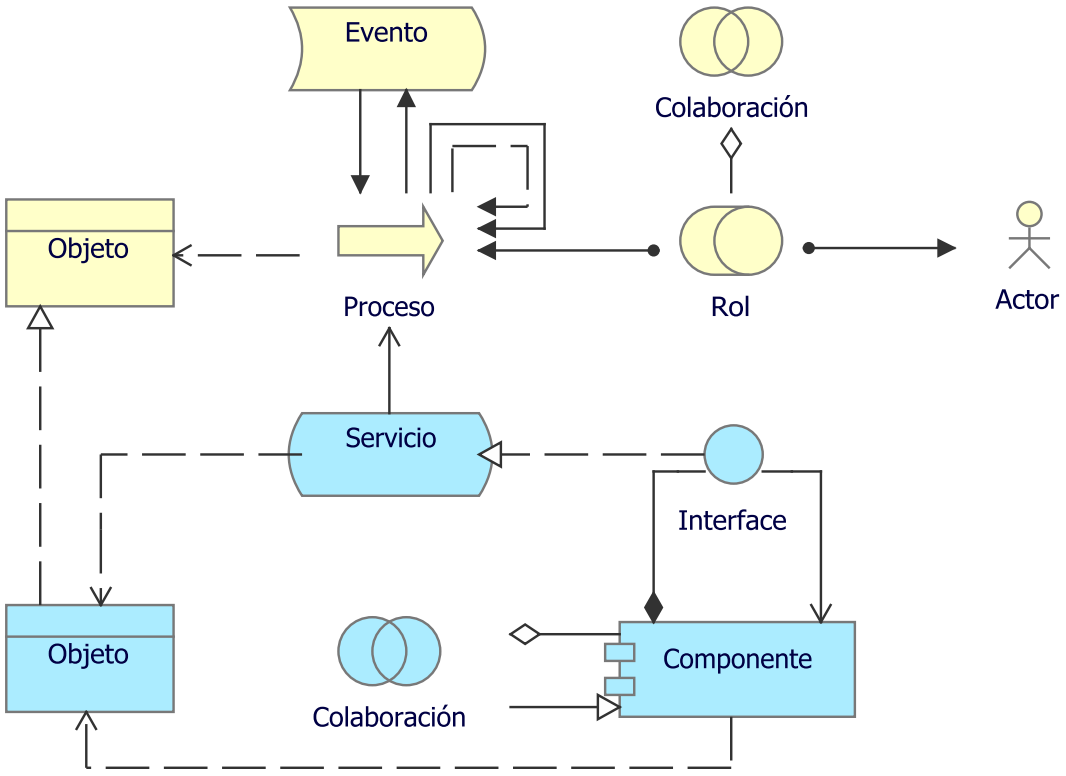


Figura 4-48.: Uso de Aplicación

Caso de uso de aplicación

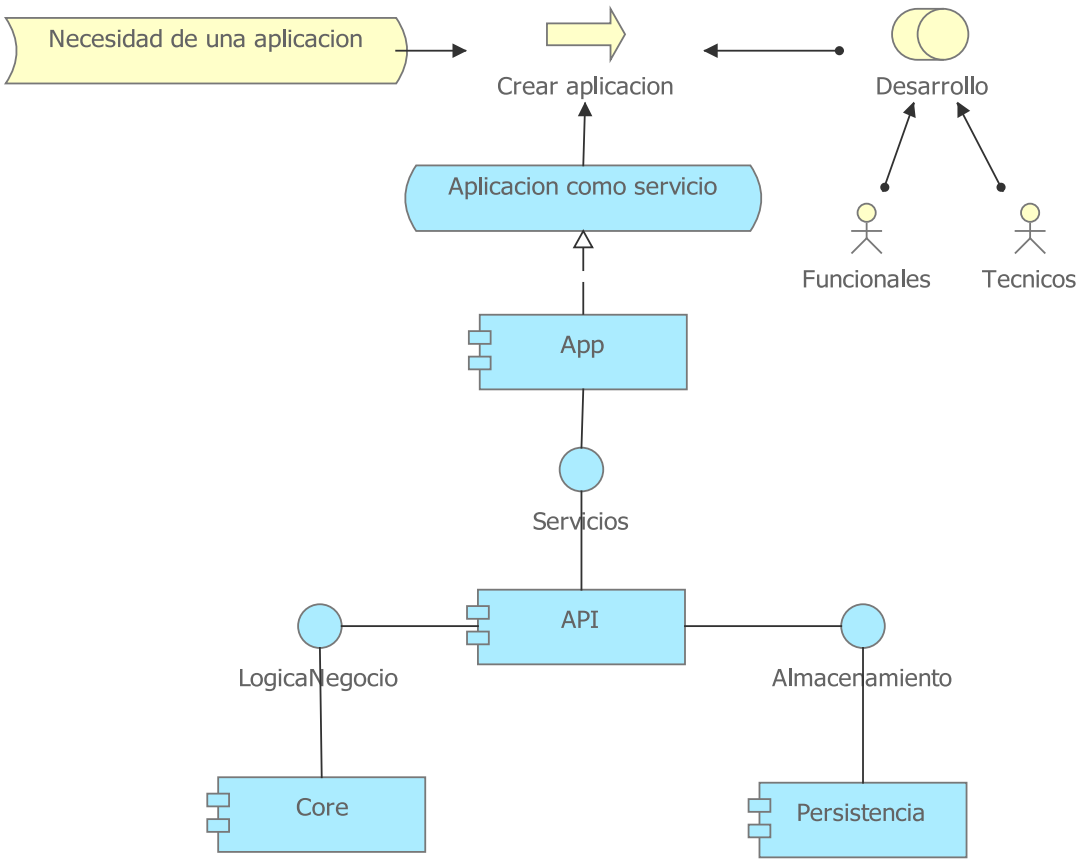


Figura 4-49.: Caso Uso de Aplicación

4.7. Capa de tecnología

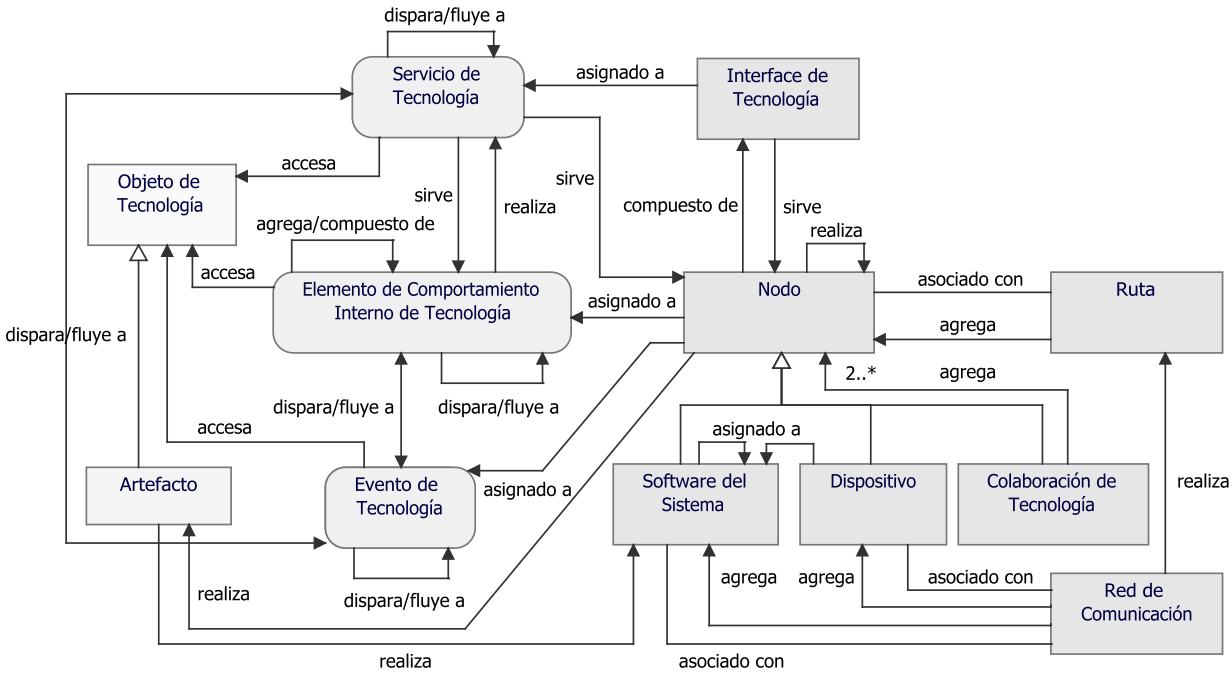


Figura 4-50.: Metamodelo de Tecnología

4.7.1. Introducción

En esta capa se establecen y dan a conocer diferentes elementos a nivel de software y hardware que dan soporte a los procesos de negocio, se usan elementos establecidos con anterioridad en las capas de negocio y aplicación, influyendo nuevos conceptos como los son Nodo y dispositivo, se busca establecer interacciones entre los diferentes elementos entre los conceptos de aplicación y negocio. [5]

4.7.2. Punto de vista de tecnología

En este punto de vista se establecen conceptos físicos que dan soporte a la aplicación, elementos como nodos y dispositivos, así como su ubicación y componentes de comunicación como el de redes. Los elementos para este punto de vista se observan en la figura **4-51**, el caso modelado propuesto se representa en la figura **4-52**

Modelo de tecnología

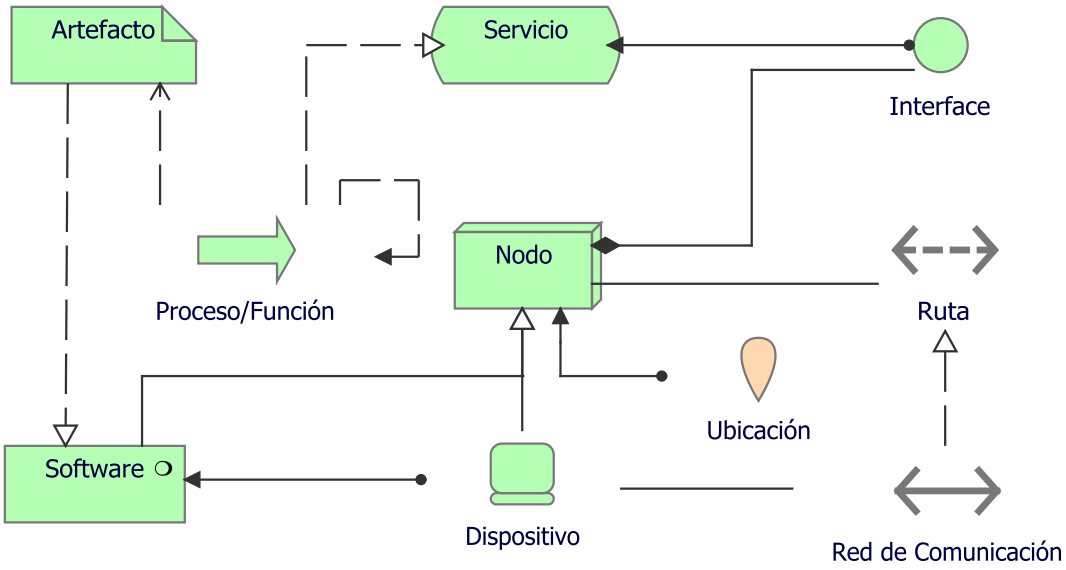


Figura 4-51.: Tecnología

Caso de tecnología

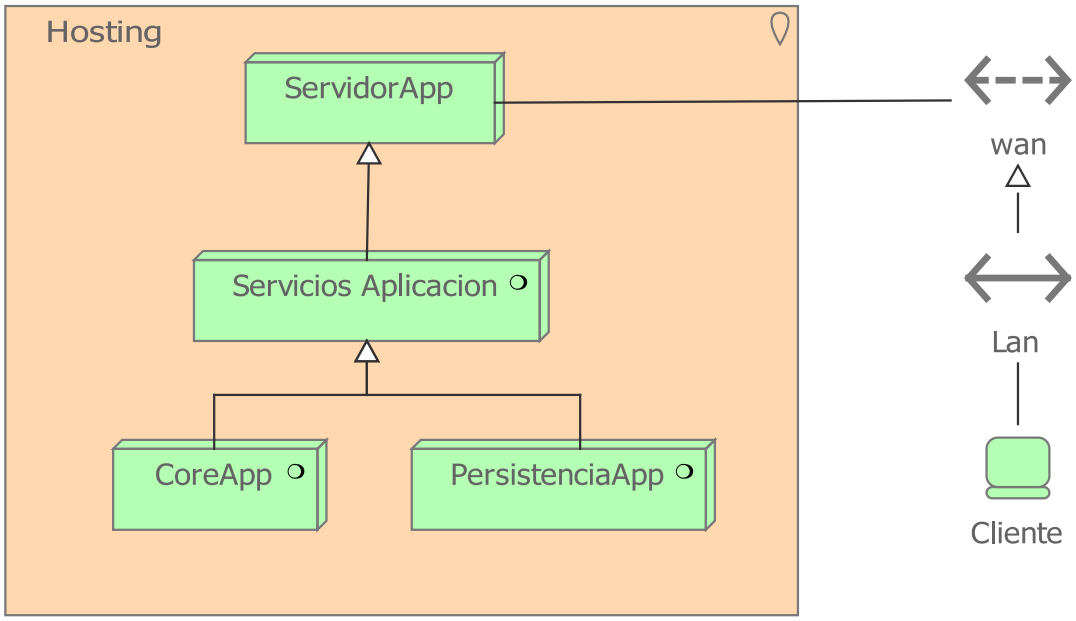


Figura 4-52.: Caso Tecnología

4.7.3. Punto de vista de uso de tecnología

Este punto de vista muestra como las aplicaciones son compatibles con la infraestructura, agregando elementos de la capa de aplicación como componente y función, según las demandas de las aplicaciones se determinan los requerimientos de la infraestructura. Los elementos para este punto de vista se observan en la figura **4-53**, el caso modelado propuesto se representa en la figura **4-54**

Modelo de uso de tecnología

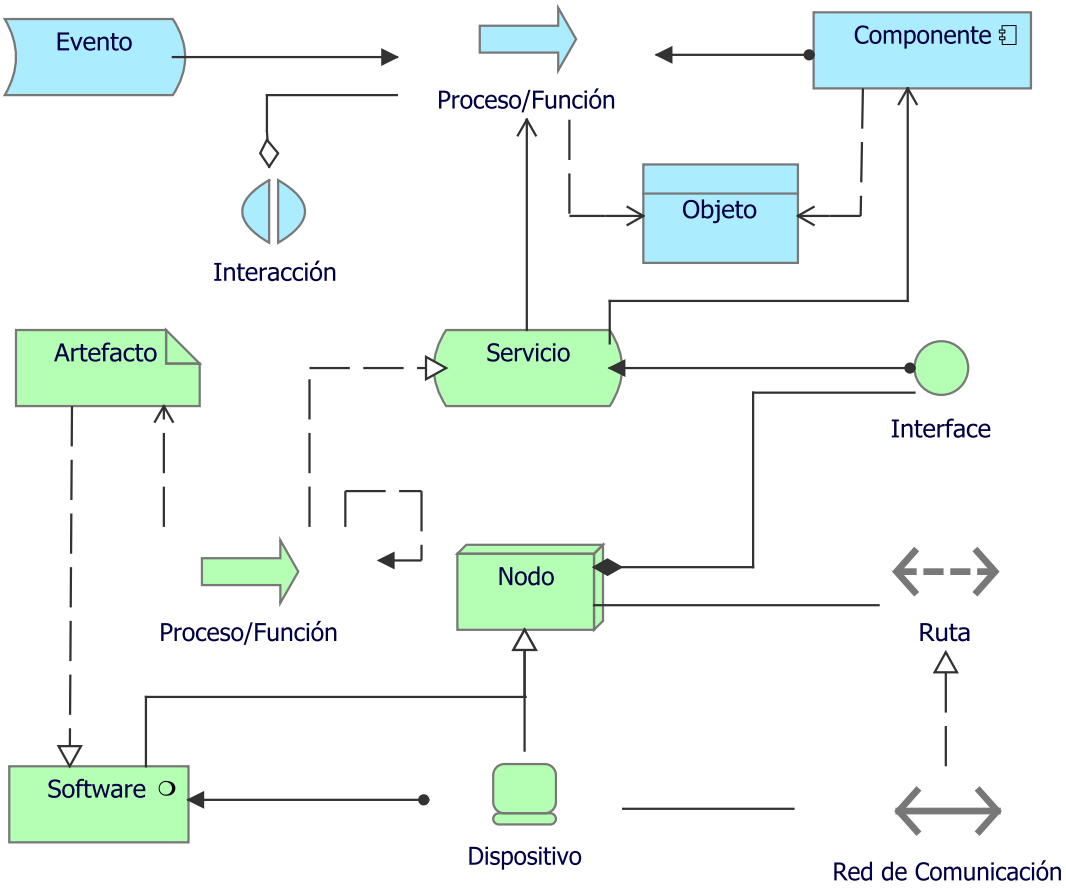


Figura 4-53.: Uso de Tecnología

Caso de uso de tecnología

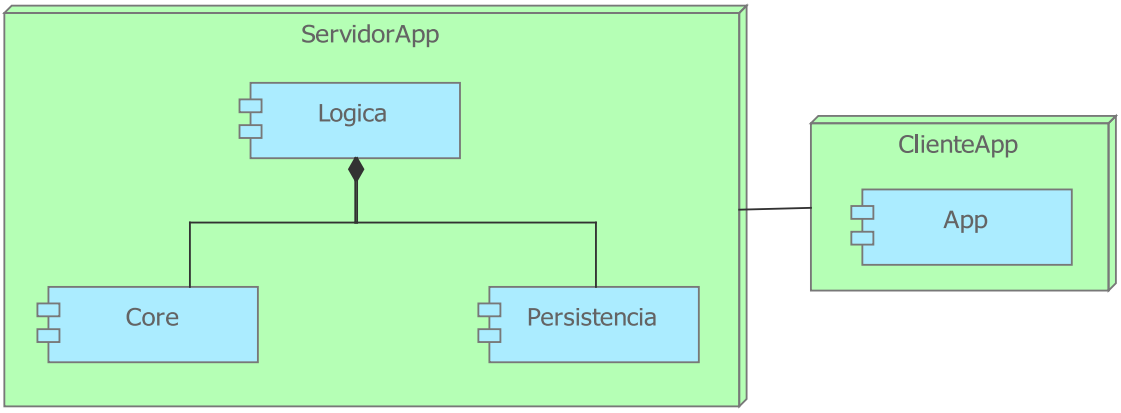


Figura 4-54.: Caso Uso de Tecnología

4.7.4. Punto de vista de despliegue e implementación

Este punto de vista permite visualizar como una aplicación se despliega en la infraestructura propuesta, agregando elementos de la capa de aplicación buscando colaboraciones estableciendo la infraestructura necesaria para las aplicaciones que soportan el negocio. Los elementos para este punto de vista se observan en la figura **4-55**, el caso modelado propuesto se representa en la figura **4-56**

Modelo de despliegue e implementación

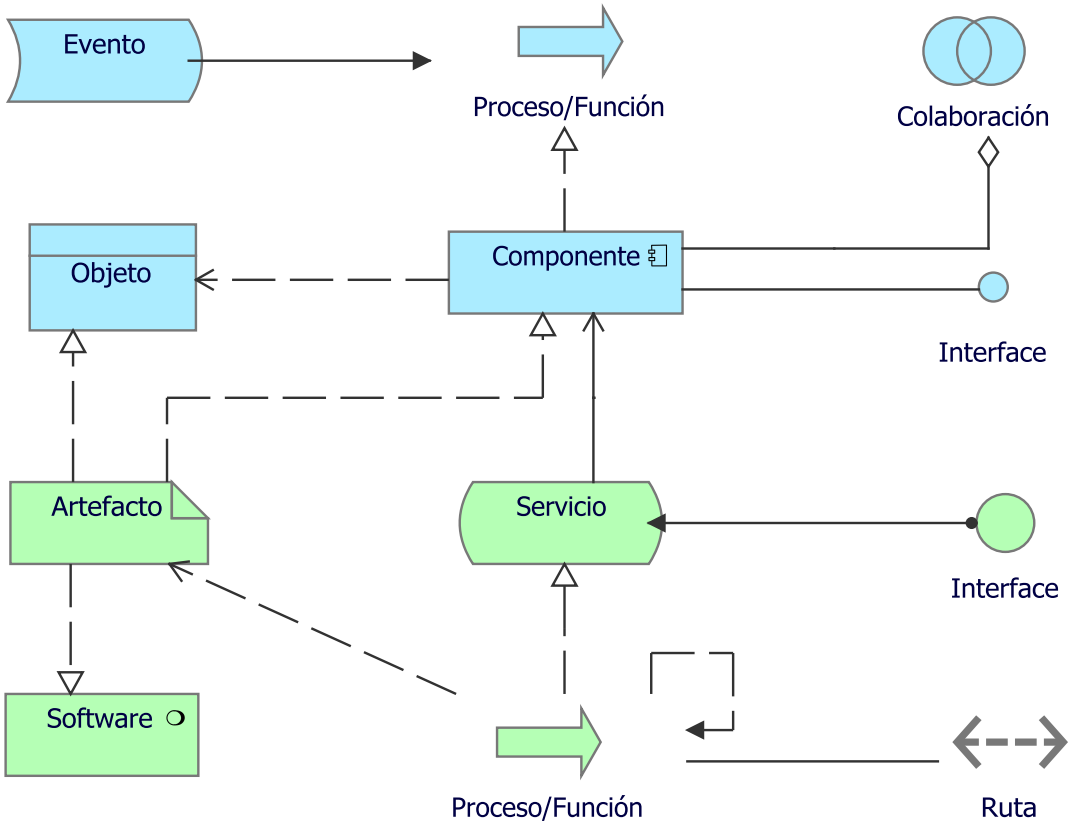


Figura 4-55.: Despliegue e Implementación

Caso de despliegue e implementación

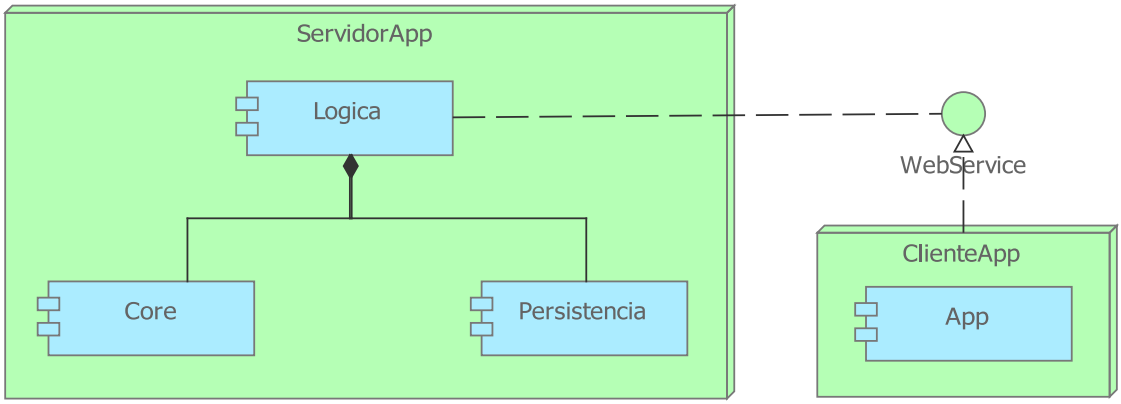


Figura 4-56.: Caso Despliegue e Implementación

4.7.5. Punto de vista de estructura de información

En este punto de vista se visualiza el modelo estructural de la información usada por las aplicaciones y dan soporte al negocio, se modelan las entidades relacionadas a modelo de datos o clases. Los elementos para este punto de vista se observan en la figura **4-57**, el caso modelado propuesto se representa en la figura **4-58**

Modelo de estructura de información

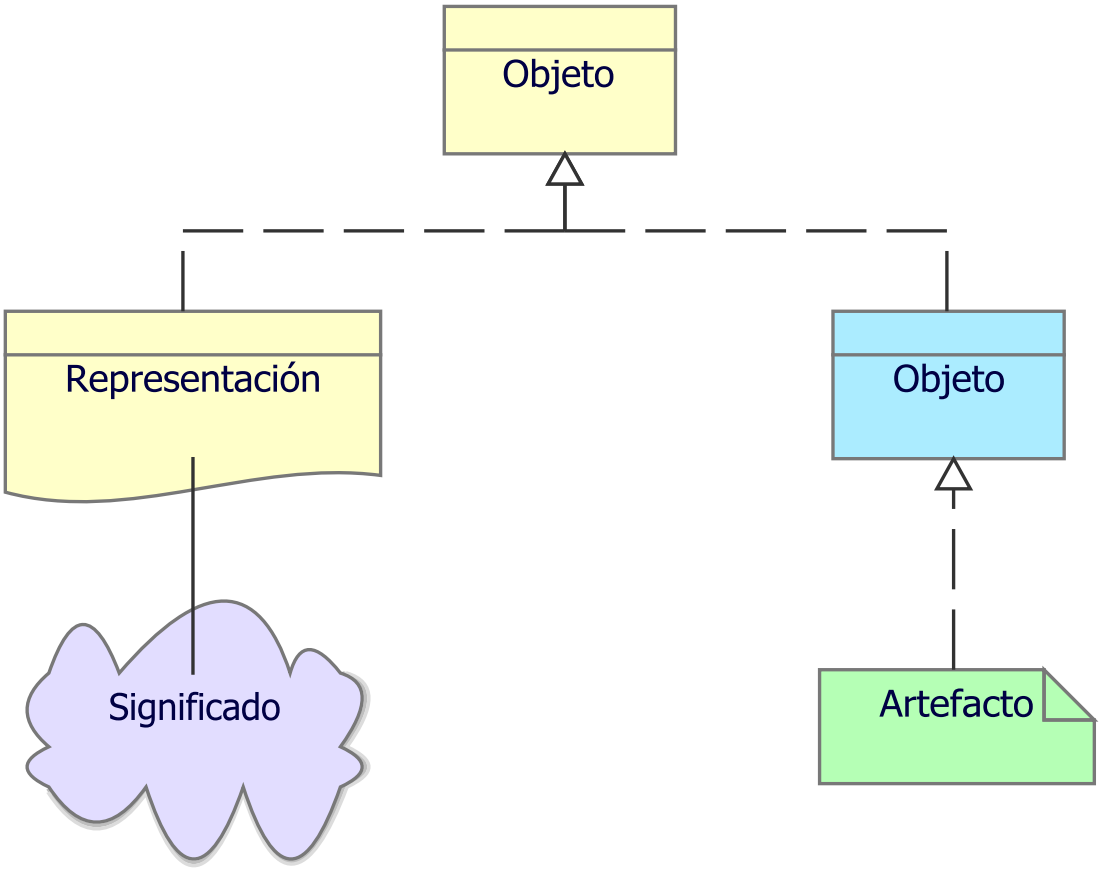


Figura 4-57.: Estructura de Información

Caso de estructura de información

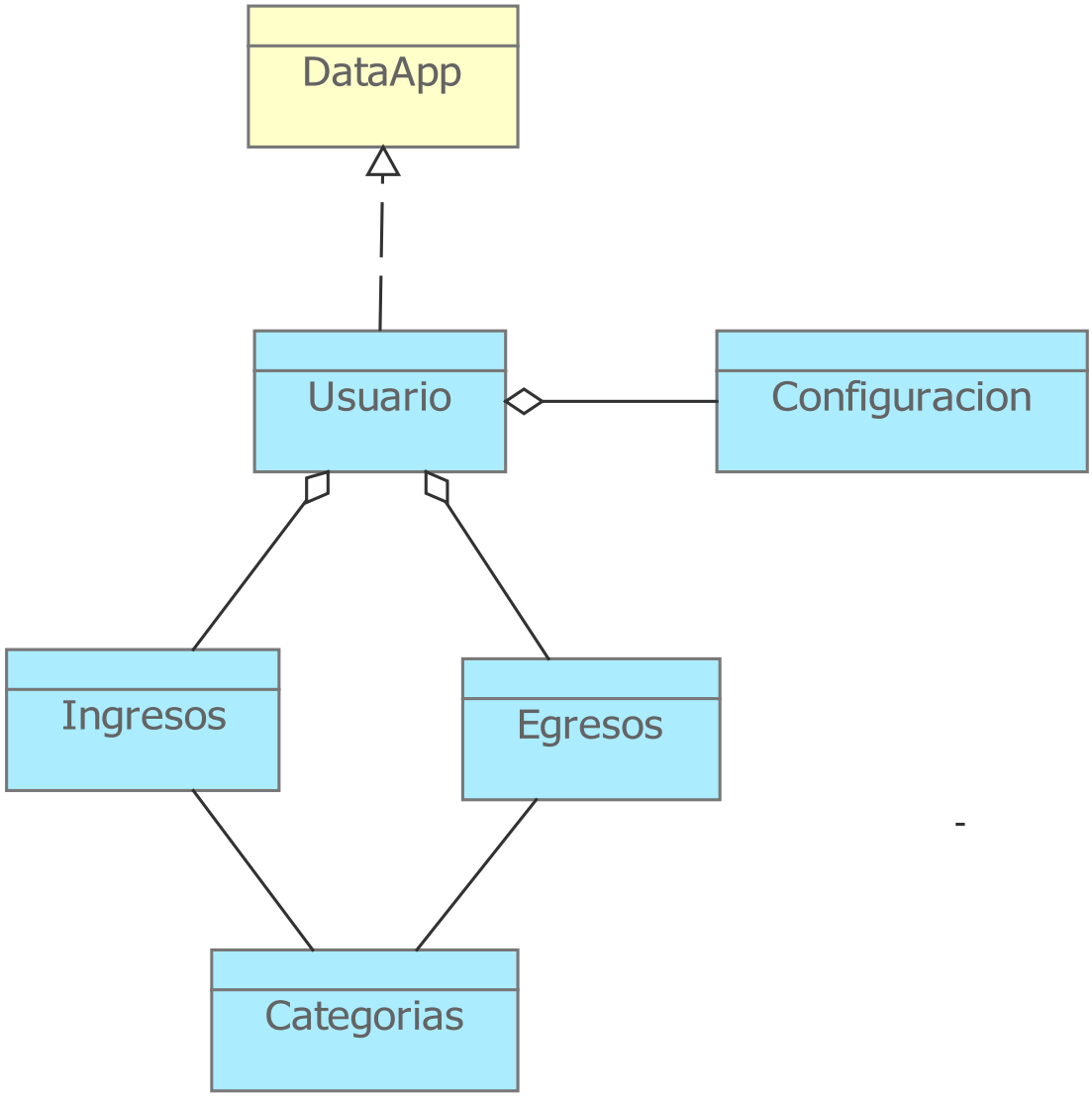


Figura 4-58.: Caso Estructura de Información

4.7.6. Punto de vista de realización del servicio

En este punto de vista se muestra como los servicios de negocio son soportados por componentes de aplicación, haciendo uso de los elementos del modelo de estructura de información uniéndolos con el negocio. Los elementos para este punto de vista se observan en la figura **4-59**, el caso modelado propuesto se representa en la figura **4-60**

Modelo de realización del servicio

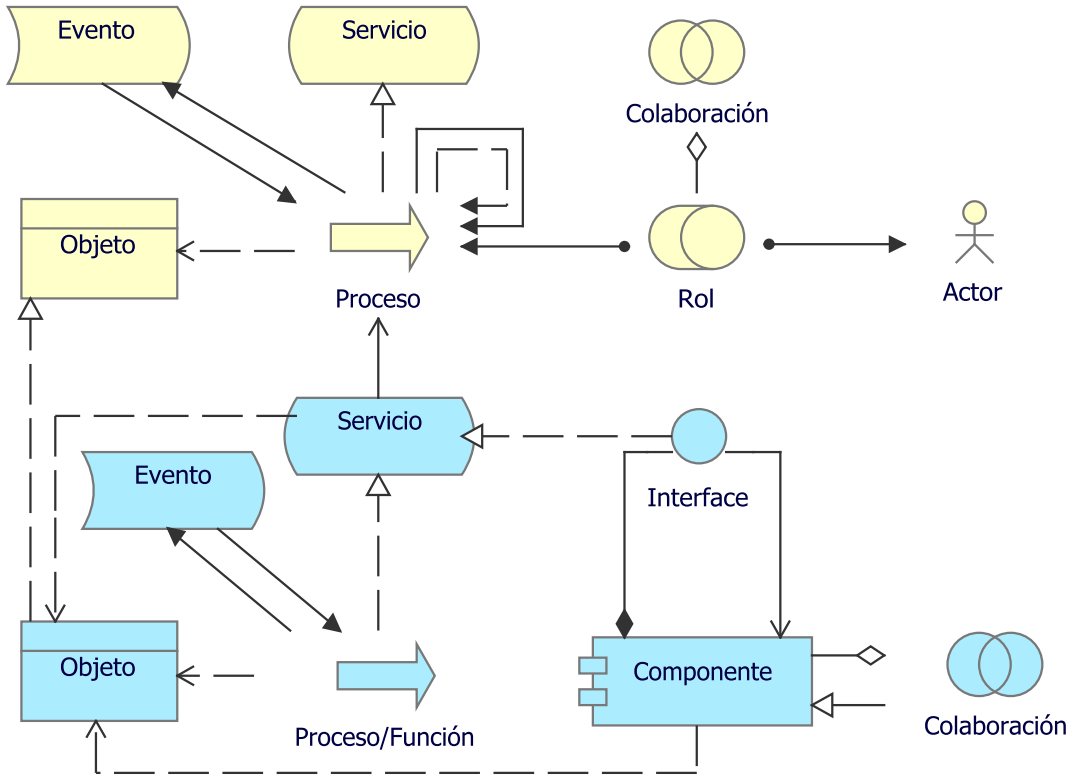


Figura 4-59.: Realización del Servicio

Caso de realización del servicio

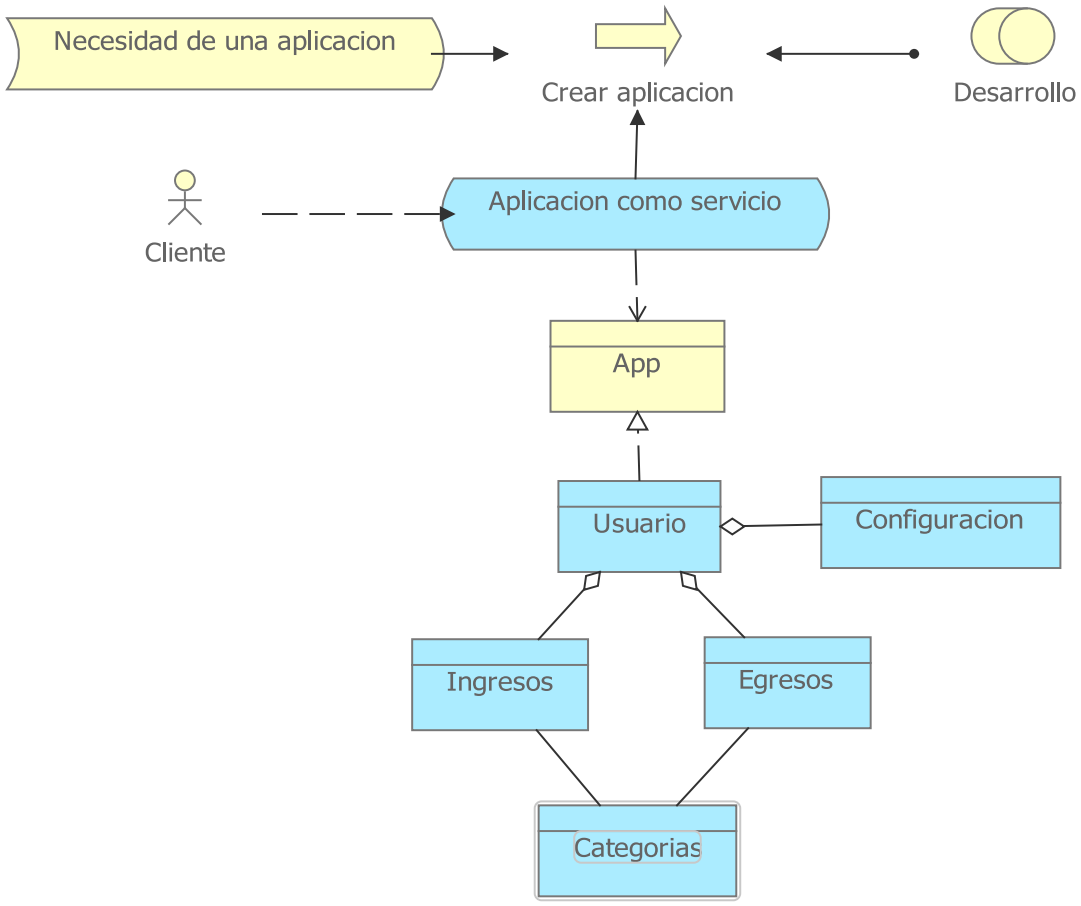


Figura 4-60.: Caso Realización del Servicio

4.7.7. Punto de vista físico

Este punto de vista se enfoca en la parte física que da soporte a los servicios del negocio, así como aspectos de ruta de distribución, esta pensado mas en servicios de tipo industrial. Los elementos para este punto de vista se observan en la figura **4-61**, el caso modelado propuesto se representa en la figura **4-62**

Modelo de físico

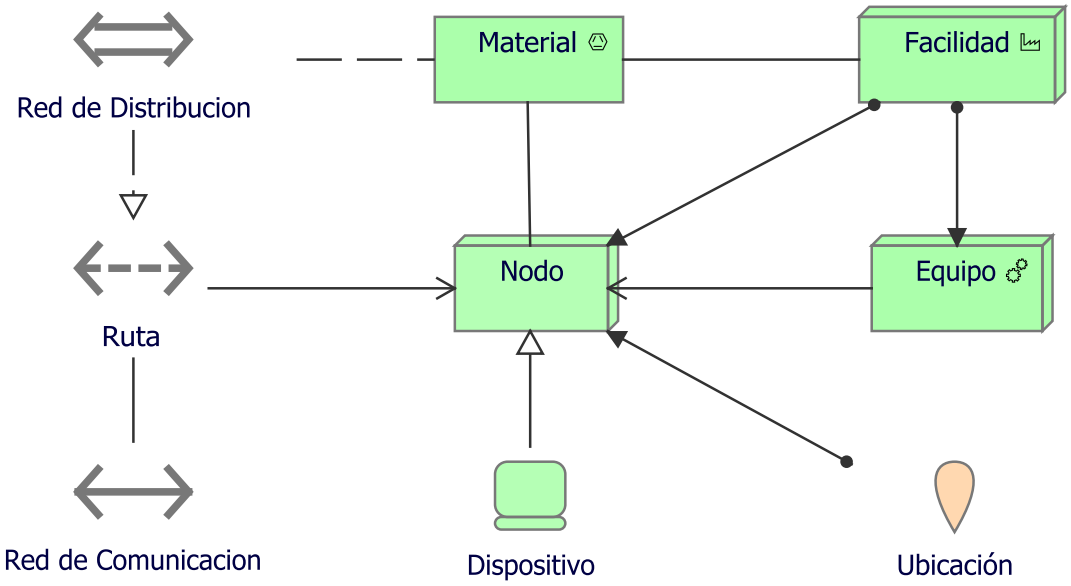


Figura 4-61.: Físico

Caso de físico



Figura 4-62.: Caso Físico

4.7.8. Punto de vista de capas

En este punto de vista se muestran las capas de negocio, aplicación y tecnología, dando una agrupación por capa a los diferentes componentes de la arquitectura de la aplicación del negocio. Los elementos para este punto de vista se observan en la figura **4-63**, el caso modelado propuesto se representa en la figura **4-64**

Modelo de capas

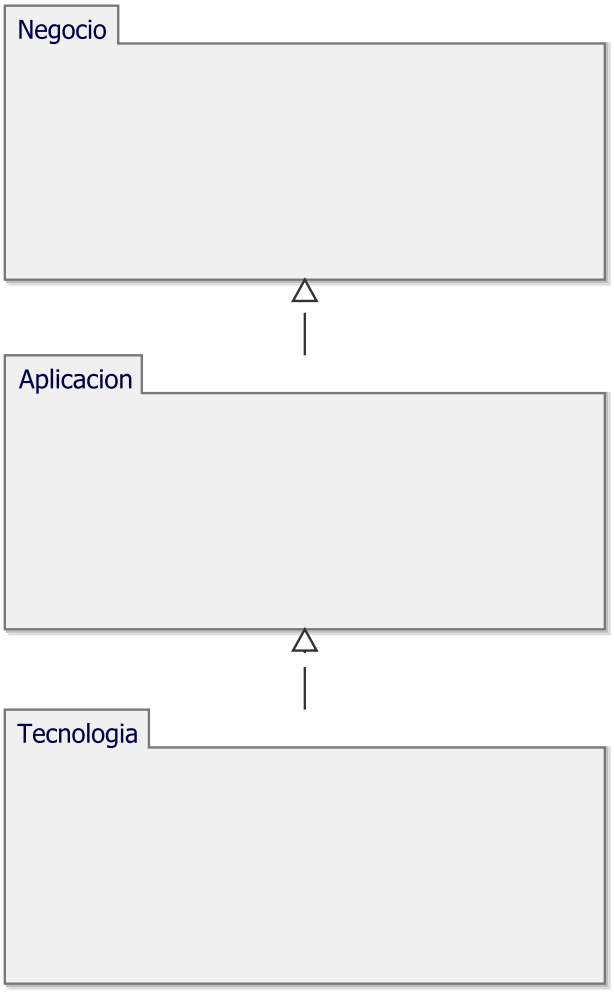


Figura 4-63.: Capas

Caso de capas

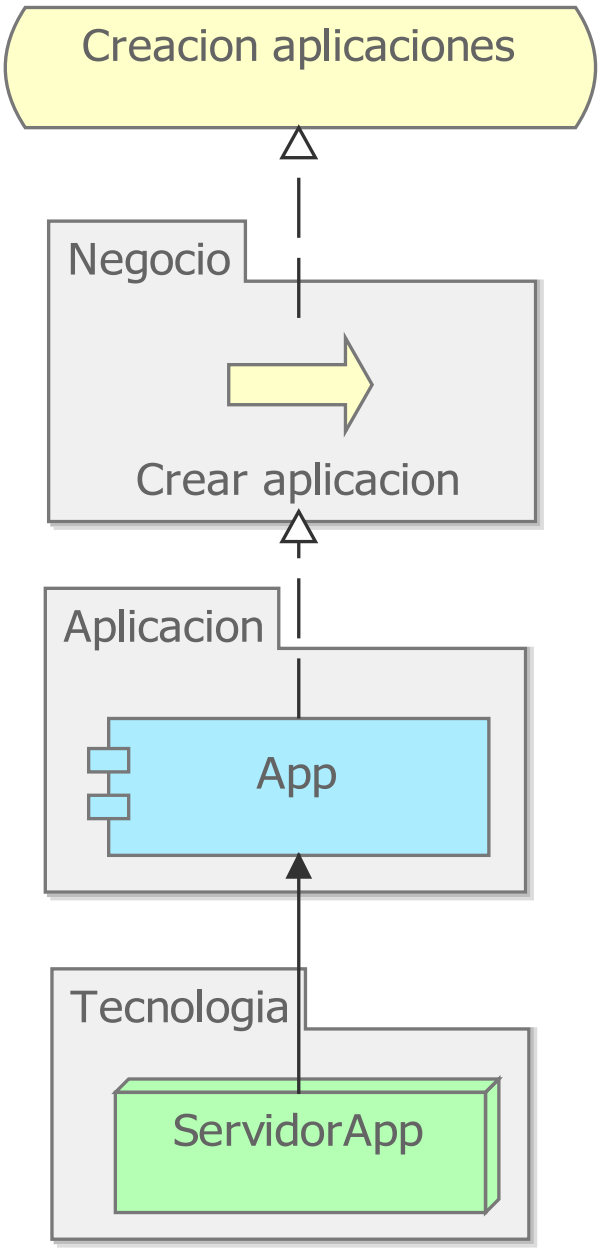


Figura 4-64.: Caso Capas

4.8. Capa de implementación y migración

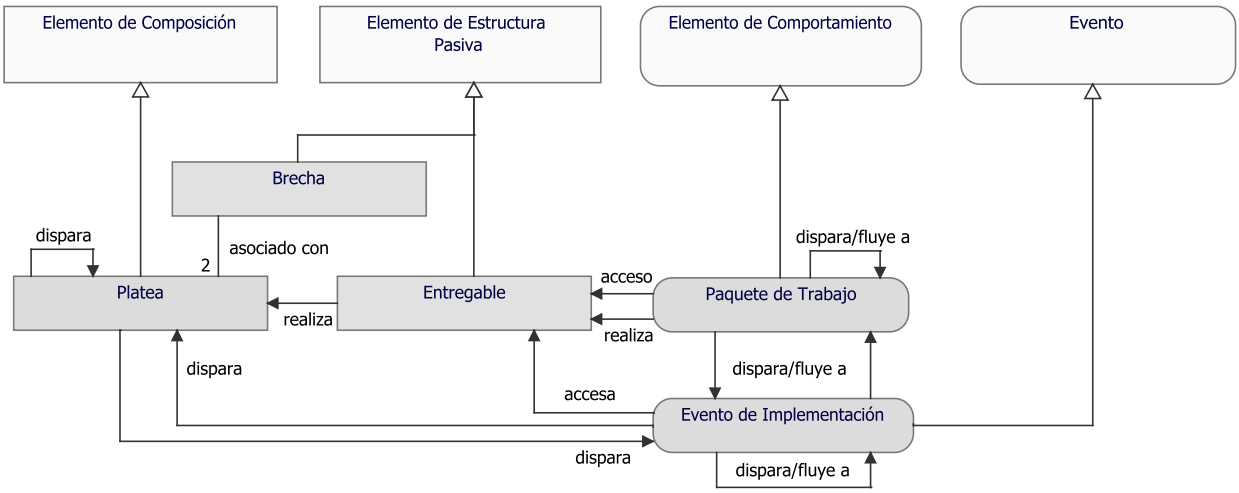


Figura 4-65.: Metamodelo de Estrategia

4.8.1. Introducción

En esta capa se exponen los modelos y características de migración para realizar un cambio en la arquitectura empresarial, presentando una perspectiva para modelar la gestión del cambio y una arquitectura de destino para la organización, dando a conocer el contexto sobre el cual se pretende modelar un cambio estructural mediante la arquitectura empresarial. En esta capa se utilizan algunos elementos de la capa de negocio empleados para crear grupos de trabajo y el concepto de liberable. [5]

4.8.2. Punto de vista de proyecto

En este punto de vista se pretende modelar la gestión de cambio de la arquitectura empresarial, gracias a paquetes de trabajo y liberable, permitiendo identificar el producto del proyecto que llevan a un objetivo empresarial y designando los roles y actores encargados. Los elementos para este punto de vista se observan en la figura **4-66**, el caso modelado propuesto se representa en la figura **4-67**

Modelo de proyecto

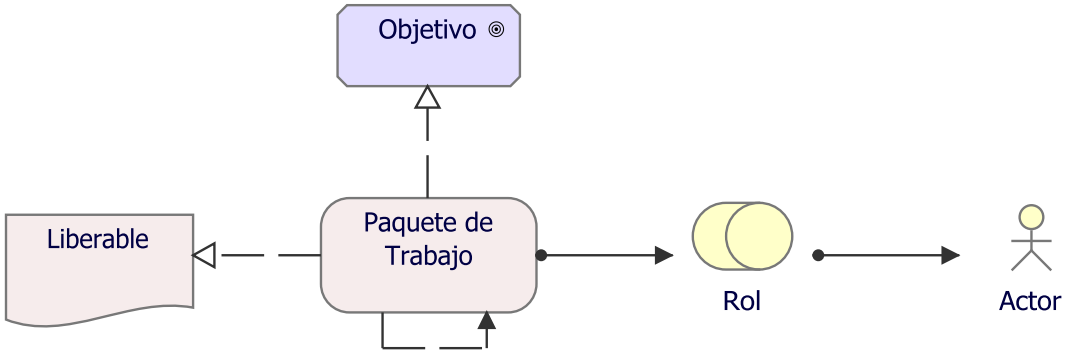


Figura 4-66.: Proyecto

Caso de proyecto

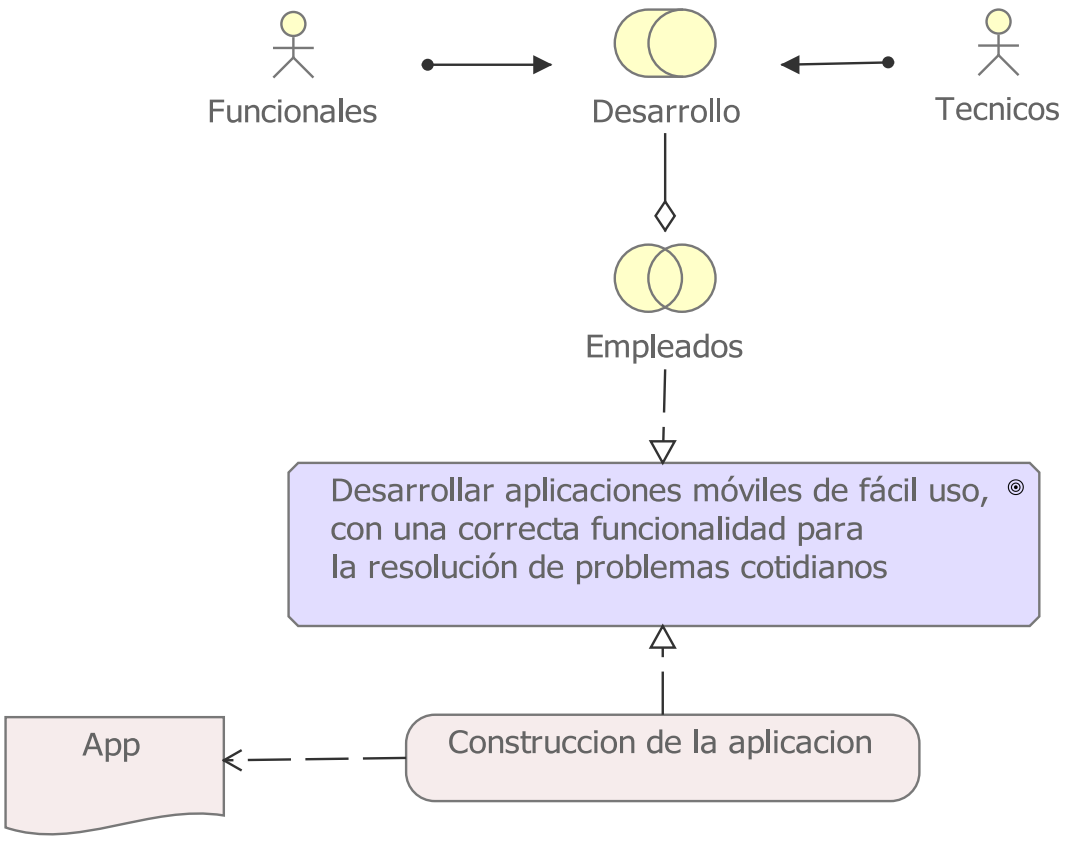


Figura 4-67.: Caso Proyecto

4.8.3. Punto de vista de migración

En este punto de vista se establecen los elementos de meseta que muestra un momento puntual del estado de desarrollo de la arquitectura y la brecha como las dificultades que se pueden presentar entre las mesetas establecidas. Los elementos para este punto de vista se observan en la figura **4-68**, el caso modelado propuesto se representa en la figura **4-69**

Modelo de migración



Figura 4-68.: Migración

Caso de migración

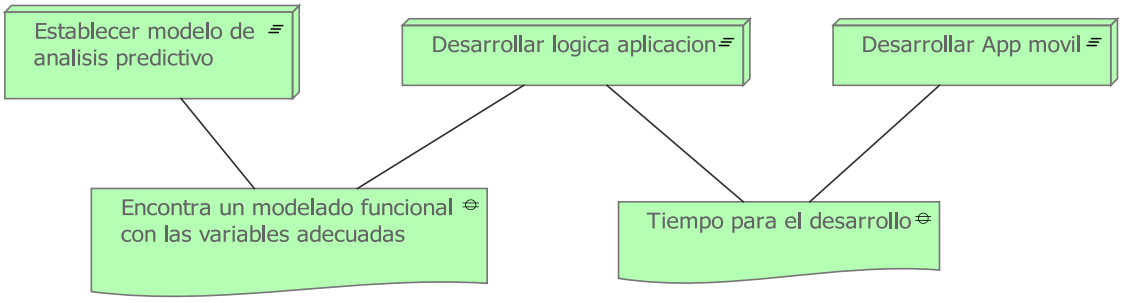


Figura 4-69.: Caso Migración

4.8.4. Punto de vista de implementación/migración

Este punto de vista integra el punto de vista de proyecto con el de migración, permitiendo relacionar los liberables en los paquetes de trabajo con las etapas del proyecto mesetas y sus brechas, relacionando también conceptos de negocio como el requerimiento y el objetivo para poder definir un modelo de apoyo para la administración de los productos. Los elementos para este punto de vista se observan en la figura **4-70**, el caso modelado propuesto se representa en la figura **4-71**

Modelo de implementación/migración

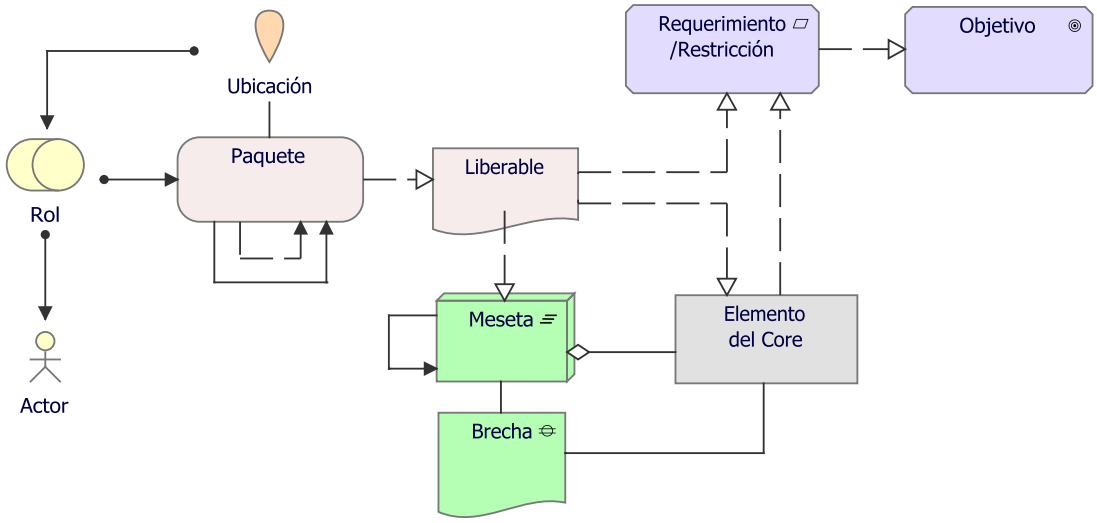


Figura 4-70.: Implementación/Migración

Caso de implementación/migración

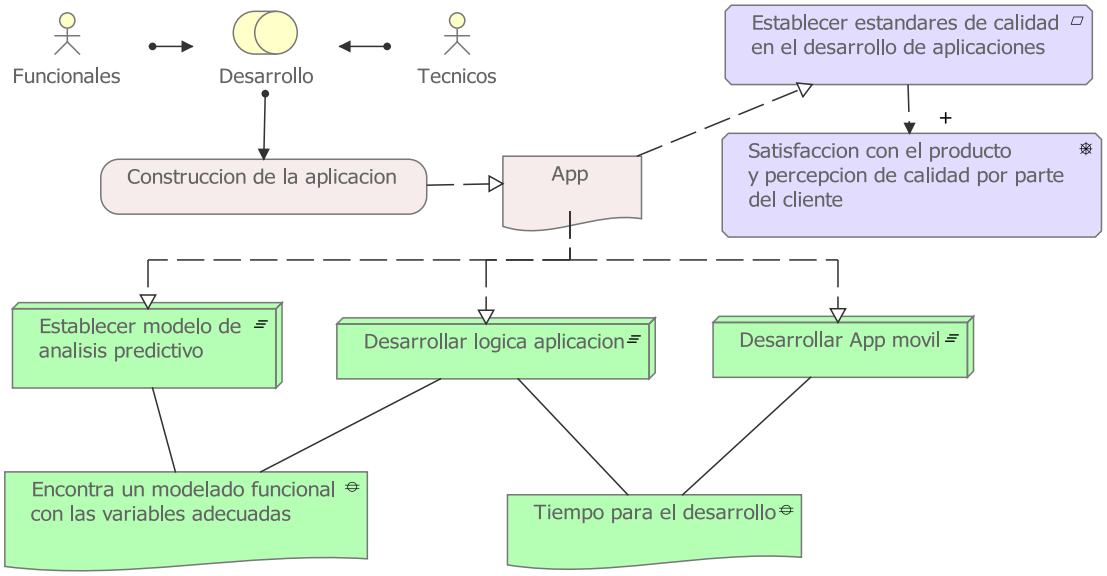


Figura 4-71.: Caso Implementación/Migración

5. PROTOTIPO DE APLICACIÓN

Para el desarrollo del prototipo de la aplicación móvil de control y gestión de finanzas personales, se eligió una arquitectura orientada a servicios SOA, creando un componente de backend que expone un API construido en Python con ayuda del framework Django Rest Framework por el cual se exponen los servicios mediante REST que serán consumidos por un cliente, en este caso una aplicación móvil construida con el framework React Native, el cual provee una aplicación híbrida basada en React y a su ves en JavaScript que puede ser desplegada tanto en Android como en IOS.

5.1. API REST (BackEnd)

El backend de la aplicación de finanzas personales se construyo en el lenguaje de programación Python, pensando en la integración con modelos de análisis predictivo, con ayuda de algunas librerías para dicho fin como pandas, numpy, seaborn, sklearn, entre otras. Siguiendo la arquitectura orientada a servicios se crea una aplicación API que expone servicios tipo REST, los cuales contienen lógica de negocio y temas de persistencia para la aplicación.
Para la creación de este API Web se utilizo Djago REST Framework, el cual provee un conjunto de herramientas para crear APIs REST de forma ágil y sencilla, con un alto grado de personalización y creación de componentes basados en vistas en los cuales se implementan operaciones CRUD de manera sencilla, provee un entorno listo para crear una aplicación que soporta un ORM para interactuar con bases de datos y paquetes como los de autenticación OAuth1a y OAuth2. [34]
La arquitectura que provee una API en Django REST Framework se muestra en la Figura **5-1** tomada de [4], en donde se muestra el ciclo de vida de las peticiones y respuestas HTTP, mediante un patrón de Url pasa a las vistas las cuales mediante un serializador interactúan con el modelo para las operaciones CRUD mediante el ORM que provee persistiendo la información en base de datos, por defecto una base de datos SQLite.
A continuacion, se detalla la estructura de la API creada, cuando se crea un proyecto con Django REST Framework se crea una estructura como la mostrada en la Figura **5-2**, en este proyecto se creo una app creando a su vez una estructura mostrada en la Figura **5-3**.

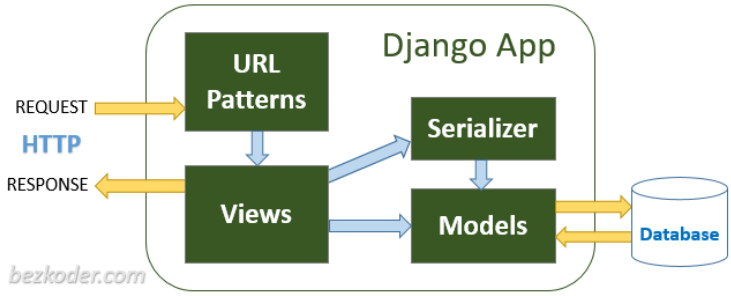


Figura 5-1.: Arquitectura Django REST Framework

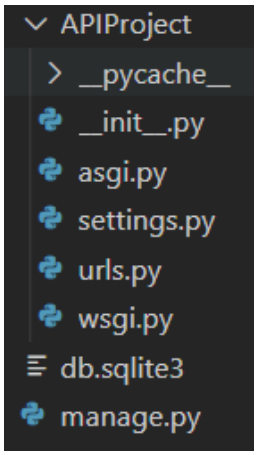


Figura 5-2.: Estructura Proyecto API

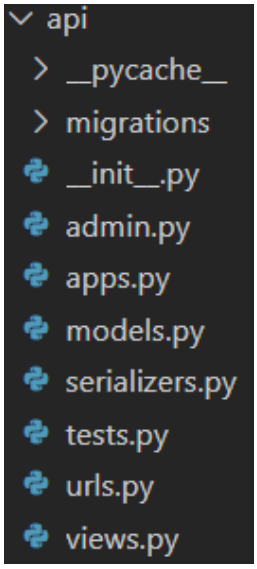


Figura 5-3.: Estructura API

Lo primero que se hizo fue crear los modelos de la aplicación, estos van a generar las tablas y relaciones correspondiente en la base de datos, los 2 modelos creados son los de Entradas para almacenar los ingresos y gastos de los usuarios con sus respectivos atributos y las categorías para estos ingresos y gastos, esto se ve en la Figura **5-4**. Después se crean los respectivos serializadores para cada Modelo Figura **5-5**, también se crean unos ViewSet con los Modelos y Serializer creaos para cada modelo, con el cual tenemos acceso al CRUD de cada Modelo Figura **5-6**, por ultimo se exponen estos ViewSet como URL para ser consumidos por REST Figura **5-7**.

models.py

api > models.py > ...

```python
from django.db import models

# Create your models here.
class Entrada(models.Model):
    valor = models.BigIntegerField()
    fecha = models.DateField(null=True)
    categoria = models.CharField(max_length=100)
    tipo = models.CharField(max_length=100)
    recurrente = models.BooleanField()

    def __str__(self):
        return self.valor

class Categoria(models.Model):
    nombre = models.CharField(max_length=100)
    tipo = models.CharField(max_length=100)

    def __str__(self):
        return self.nombre

```

Figura 5-4.: API - Modelo

serializers.py

```python
from rest_framework import serializers
from .models import Entrada
from .models import Categoria

class EntradaSerializer(serializers.ModelSerializer):
    class Meta:
        model = Entrada
        fields = ['id', 'valor', 'fecha', 'categoria', 'tipo', 'recurrente']

class CategoriaSerializer(serializers.ModelSerializer):
    class Meta:
        model = Categoria
        fields = ['id', 'nombre', 'tipo']
```

Figura 5-5.: API - Serializer

views.py

```python
from django_filters.rest_framework import DjangoFilterBackend
from rest_framework.filters import SearchFilter, OrderingFilter

class EntradaViewSet(viewsets.ModelViewSet):
    queryset = Entrada.objects.all()
    serializer_class = EntradaSerializer
    filter_backends = [DjangoFilterBackend, SearchFilter, OrderingFilter]
    filterset_fields = ['id', 'tipo']
    search_fields = ['=tipo', 'categoria']
    ordering_fields = ['tipo', 'fecha']
    ordering = ['id']

class CategoriaViewSet(viewsets.ModelViewSet):
    queryset = Categoria.objects.all()
    serializer_class = CategoriaSerializer
    filter_backends = [DjangoFilterBackend, SearchFilter, OrderingFilter]
    filterset_fields = ['id', 'tipo']
    search_fields = ['=tipo', 'nombre']
    ordering_fields = ['tipo']
    ordering = ['id']
```

Figura 5-6.: API - ViewSet

Cuando ponemos a correr el proyecto API REST, Django provee un CRUD completo para interactuar con los modelos definidos, mediante URL podemos Listar todos o un registro, filtrarlos por los campos configurados en el ViewSet, así como Crear, Modificar o Eliminar registros, accediendo a la URL en donde corre la aplicacion se despliega una interfaz grafica para interactuar directamente con los registros, Figura **5-8**, o en el caso de la APP móvil accediendo mediante peticiones REST por el protocolo HTTP.

urls.py

api > urls.py > ...

```python
from django.urls import path
from .views import EntradaViewSet
from .views import CategoriaViewSet
from rest_framework.routers import DefaultRouter
from django.conf.urls import include

router = DefaultRouter()
router.register('entradas', EntradaViewSet, basename='entradas')
router.register('categorias', CategoriaViewSet, basename='categorias')

urlpatterns = [
    path('api/', include(router.urls)),

]
```

Figura 5-7.: API - URL

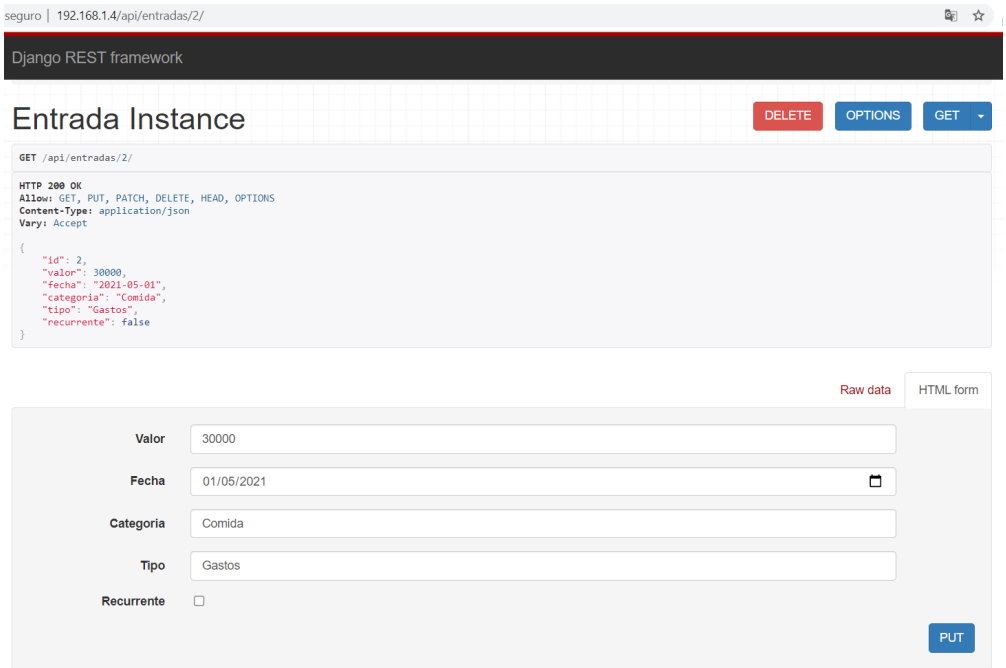


Figura 5-8.: CRUD Django REST API

5.2. APP Móvil (FrontEnd)

Para la parte de la aplicación móvil de la aplicación de finanzas personales, se utilizo React Native, para construir una aplicación que se comporta de forma nativa para Android y IOS. React Native es un framework de React a su vez basado en JavaScript, con el cual se construyen interfaces de usuario para móviles, con elementos de desarrollo web como JavaScript, CSS y etiquetas, renderizando los elementos a aplicaciones nativas para Anroid

y IOS, permitiendo el desarrollo simultaneo para estas plataformas. [17] React Native puede ejecutarse en un navegador web y en dispositivos móviles directamente, la arquitectura puede apreciarse en la Figura **5-9** tomada de [14].

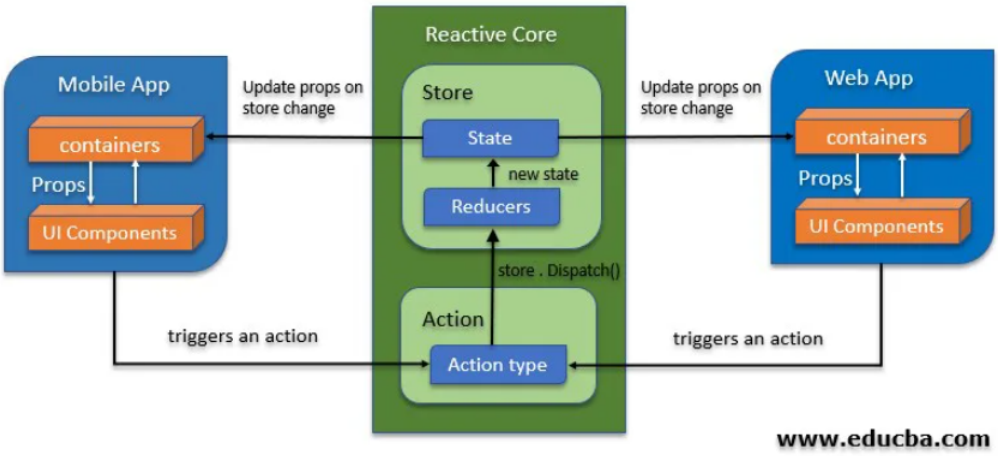


Figura 5-9.: Arquitectura React Native

La estructura básica de la aplicación construida con React Native se representa en la Figura **5-10**, se utiliza Node para gestionar paquetes necesarios para la construcción de la aplicación, se utilizaron varios paquetes para crear componentes de interfaz de usuario, estilos y funcionalidades como de navegación entre pantallas, en la Figura **5-11** se aprecian las diferentes pantallas de navegación de la aplicación creada.

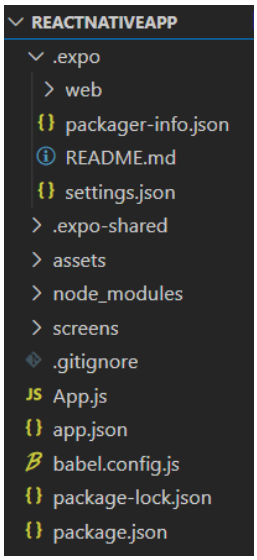


Figura 5-10.: Estructura React Native

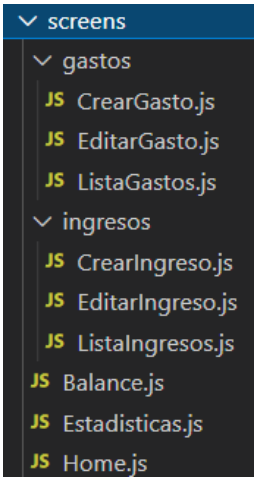


Figura 5-11.: Pantallas React Native

El archivo de inicio de ejecución de la aplicación es el App,js, en este se creó un estilo de navegación entre pantallas llamado StackNavigator del componente importado React Navigation, como se aprecia en la Figura **5-12**. La pantalla principal de la aplicación es la de Home.js, en esta se crea el menú de la aplicación móvil, la cual consta de 4 componentes principales: Ingresos, Gastos, Balance y Estadísticas, el código se aprecia en la Figura **5-13** y la vista principal en la aplicación se aprecia en la Figura **5-14**.

App.js

```
function App() {
  return (
    <View style={styles.container}>
      <Stack.Navigator>
        <Stack.Screen name = "Home" component = {Home} options = {navigationStyles} />
        <Stack.Screen name = "ListaGastos" component = {ListaGastos} options = {{...navigationStyles, title: "Lista de gastos"}}/>
        <Stack.Screen name = "CrearGasto" component = {CrearGasto} options = {{...navigationStyles, title: "Nuevo gasto"}}/>
        <Stack.Screen name = "EditarGasto" component = {EditarGasto} options = {{...navigationStyles, title: "Editar gasto"}}/>
        <Stack.Screen name = "ListaIngresos" component = {ListaIngresos} options = {{...navigationStyles, title: "Lista de ingresos"}}/>
        <Stack.Screen name = "CrearIngreso" component = {CrearIngreso} options = {{...navigationStyles, title: "Nuevo ingreso"}}/>
        <Stack.Screen name = "EditarIngreso" component = {EditarIngreso} options = {{...navigationStyles, title: "Editar ingreso"}}/>
        <Stack.Screen name = "Balance" component = {Balance} options = {{...navigationStyles, title: "Balance"}}/>
        <Stack.Screen name = "Estadisticas" component = {Estadisticas} options = {{...navigationStyles, title: "Estadisticas"}}/>
      </Stack.Navigator>
    </View>
  );
}

export default () => {
  return (
    <NavigationContainer>
      <App/>
    </NavigationContainer>
  )
}
```

Figura 5-12.: App.js - StackNavigator

Home.js

screens > Home.js > Home

```
17      }
18
19      return (
20          <View style={{flex:1}}>
21              <Button style = {styles.buttonStyle}
22                  icon = "bookmark-plus-outline"
23                  mode = "contained"
24                  color = "green"
25                  onPress = {() => props.navigation.navigate("ListaIngresos", {configData:configData})}
26              >Ingresos</Button>
27              <Button style = {styles.buttonStyle}
28                  icon = "bookmark-minus-outline"
29                  mode = "contained"
30                  color = "red"
31                  onPress = {() => props.navigation.navigate("ListaGastos", {configData:configData})}
32              >Gastos</Button>
33              <Button style = {styles.buttonStyle}
34                  icon = "scale-balance"
35                  mode = "contained"
36                  color = "blue"
37                  onPress = {() => props.navigation.navigate("Balance", {configData:configData})}
38              >Balance</Button>
39              <Button style = {styles.buttonStyle}
40                  icon = "chart-bar"
41                  mode = "contained"
42                  color = "purple"
43                  onPress = {() => props.navigation.navigate("Estadisticas", {configData:configData})}
44              >Estadisticas</Button>
45          </View>
```

Figura 5-13.: Home.js - Menú principal

5.2.1. Componentes ingresos y gastos

Mediante los componentes de Ingresos y Gastos, el usuario puede ingresar y gestionar sus ingresos y gastos personales, estos componentes de la aplicación móvil consumen los servicios expuestos por el API creado en Django Rest Framework, permitiendo consultar, crear, actualizar y eliminar registros mediante consumos REST, un ejemplo de consumo desde la aplicación se aprecia en la Figura **5-15**. En la aplicación se puede establecer para los ingresos y gastos el valor, una categoría que varia según si es ingreso o gasto, establecer si es algo recurrente que se vera reflejado cada mes o si no es recurrente seleccionar una fecha especifica. El componente de listar Ingresos se aprecia en la Figura **5-16** y el de listar gastos en la Figura **5-17**. El proceso de creación, edición o eliminación de gastos e ingresos se pueden observar en la Figura **5-17** y en la Figura **5-18**.

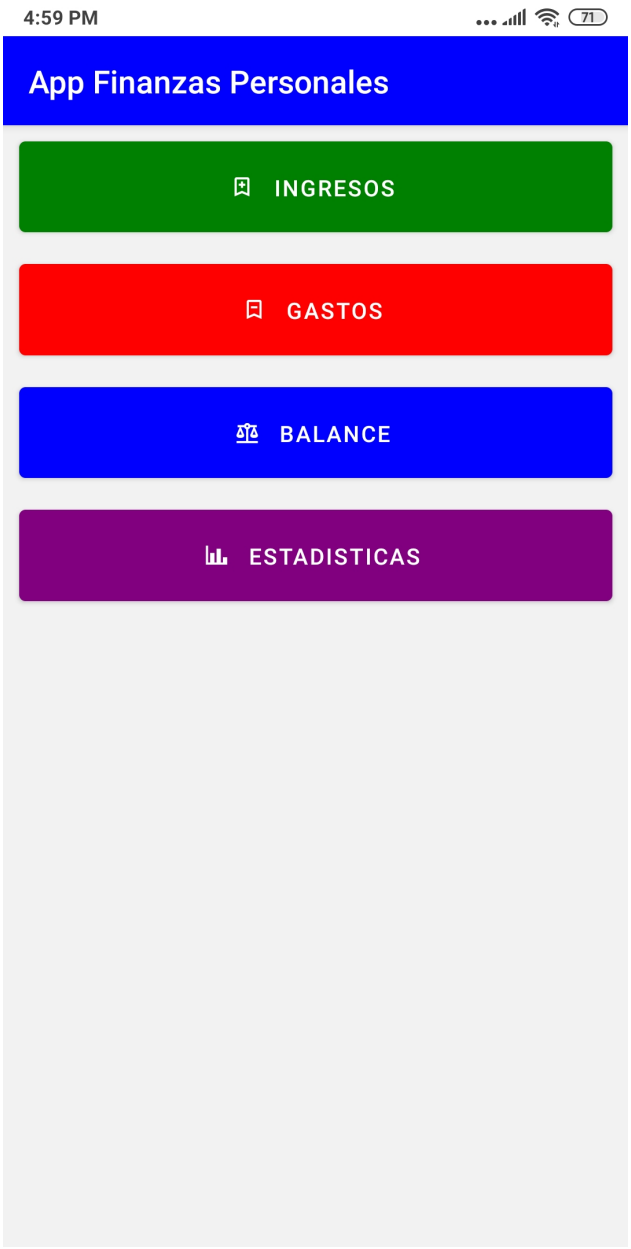


Figura 5-14.: App - Menú principal

5.2.2. Componente balance

Mediante el componente de Balance, el usuario puede visualizar los ingresos y gastos realizados en el mes actual así como el total de Ingresos, Gastos y el saldo disponible para el mes en curso, como se aprecia en la Figura **5-20**.

JS ListaGastos.js ×

screens > gastos > JS ListaGastos.js > ListaGastos > useEffect() callback

```
1  import React, {useState, useEffect} from 'react'
2  import { StyleSheet, Text, View, FlatList, Alert } from 'react-native';
3  import {Card, FAB} from 'react-native-paper';
4  import numbro from "numbro";
5
6  function ListaGastos(props) {
7      const configData = props.route.params.configData;
8      const [data, setData] = useState([])
9      const [loading, setLoading] = useState(true)
10
11     const loadData = () => {
12         fetch(configData.urlService + '/entradas/?tipo=Gasto&ordering=fecha', {
13             method:"GET"
14         })
15         .then(resp => resp.json())
16         .then(data => {
17             setData(data)
18             setLoading(false)
19         })
20         .catch(error => Alert.alert("error", error))
21     }
22
23     useEffect(() => {
24         loadData();
25     }, [])
26
27     const clickedItem = (data) => {
28         props.navigation.navigate("EditarGasto", {data:data, configData:configData})
29     }
```

Figura 5-15.: App consumo servicio listar Ingresos

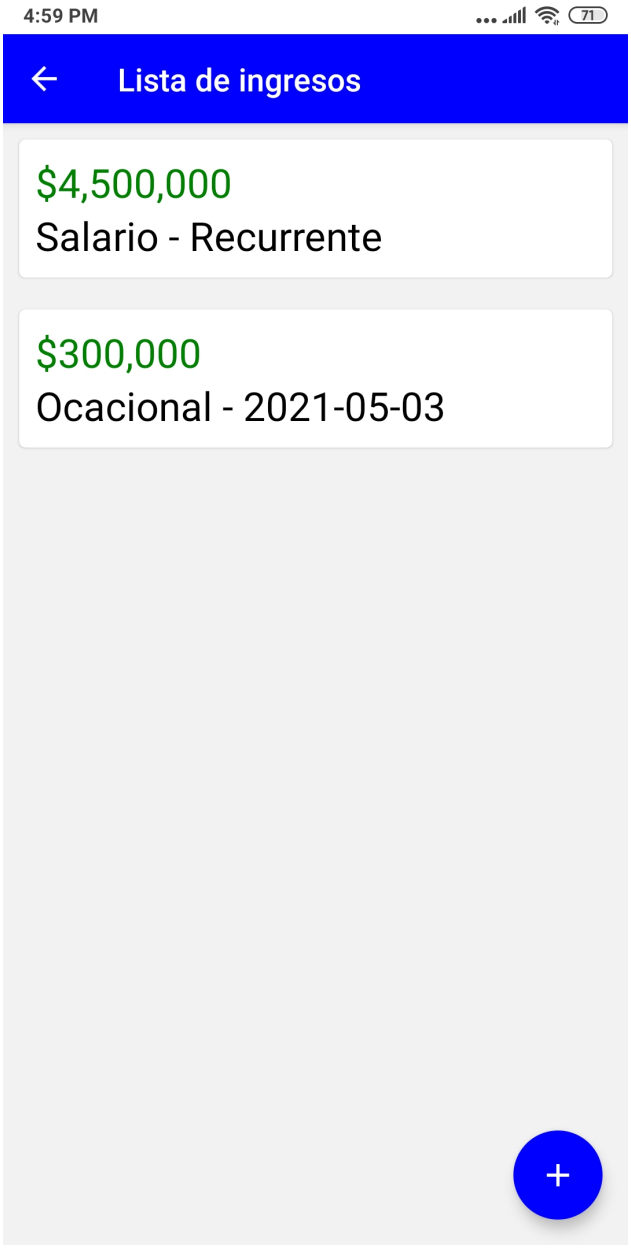


Figura 5-16.: App - Listar ingresos

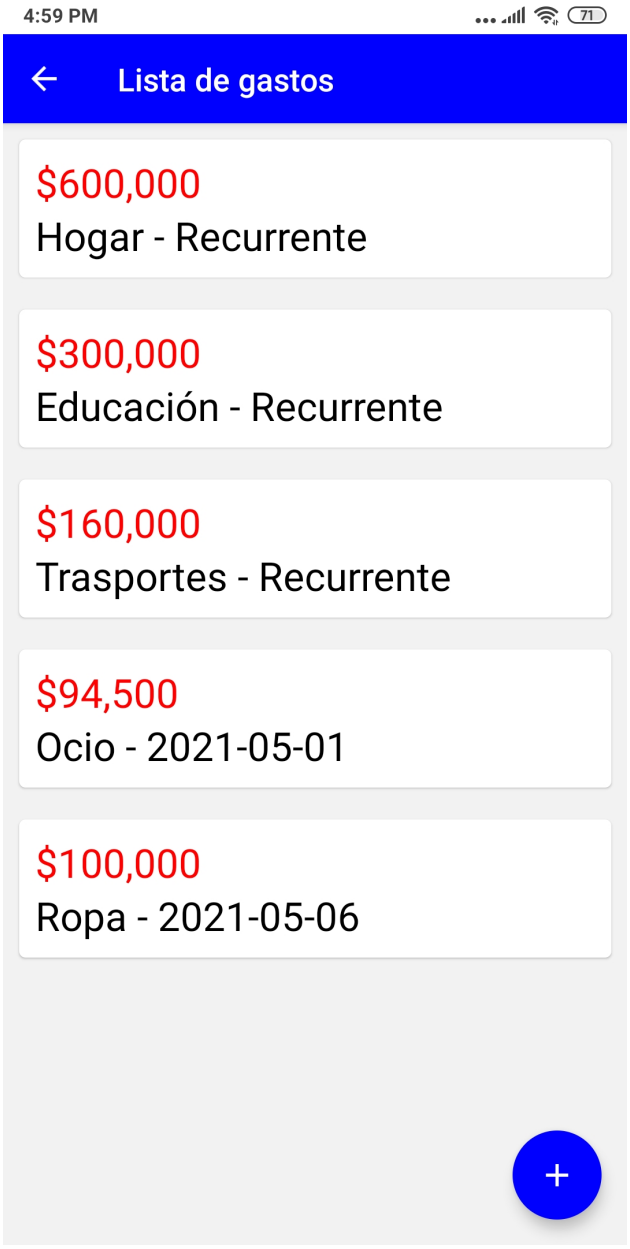


Figura 5-17.: App - Listar gastos

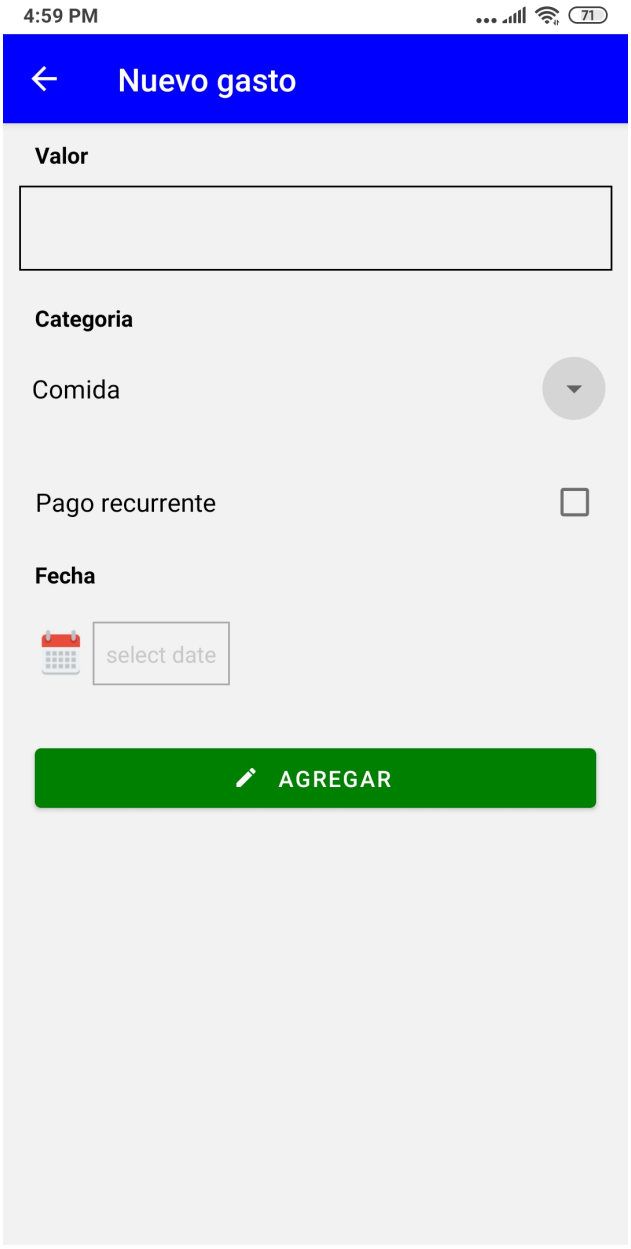


Figura 5-18.: App - Crear gasto

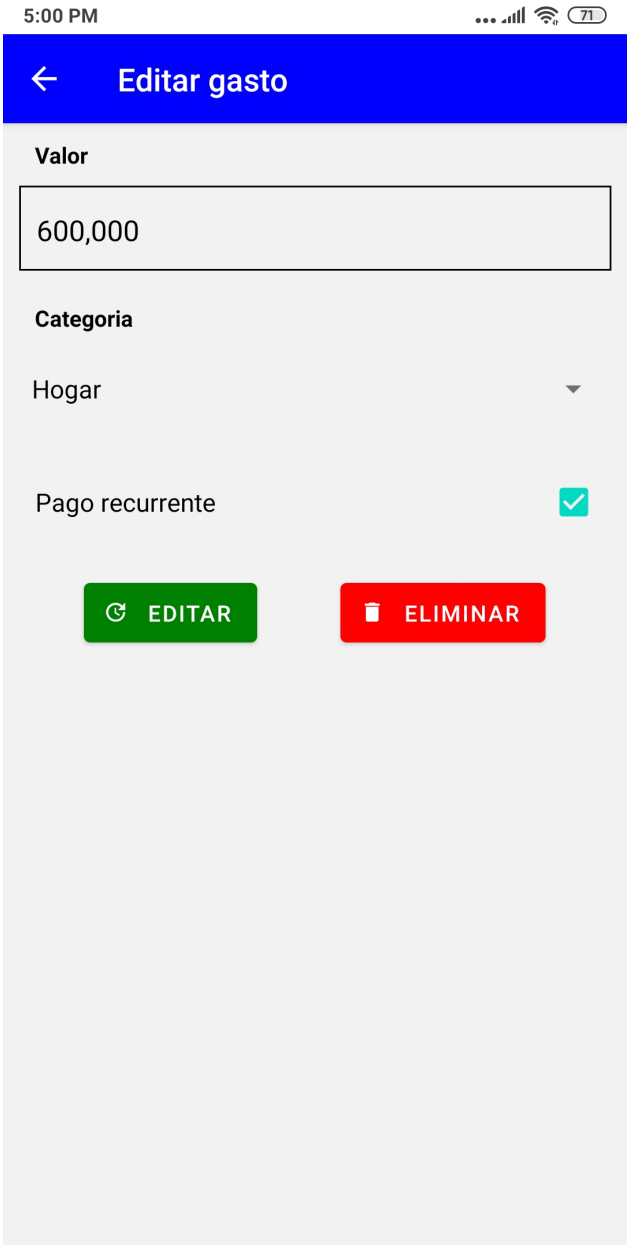


Figura 5-19.: App - Editar gasto



Figura 5-20.: App - Balance

5.2.3. Componente estadísticas

Mediante el componente de Estadísticas el usuario encuentra gráficos estadísticos útiles productos del análisis de los datos ingresados en la aplicación, mediante estos gráficos el usuario puede visualizar su comportamiento financiero, llevando un control visual y estadístico de los gastos realizados, en este componente se utiliza el modelo de análisis predictivo creado a partir de las técnicas de machine learning, mediante el cual se pueden crear gráficos que orientan al usuario acerca de su comportamiento financiero, llevándolo a la toma de decisiones en mejora de sus finanzas personales, el aspecto de este modulo se aprecia en la Figura **5-21**.

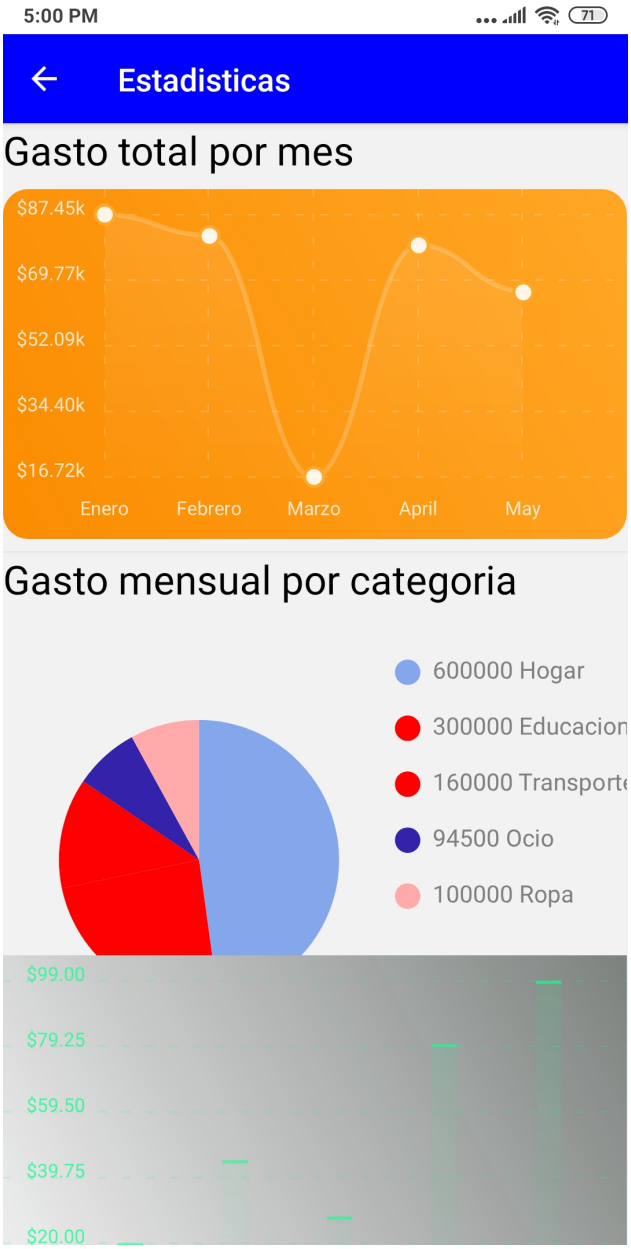


Figura 5-21.: App - Estadísticas

5.3. Base de datos

La base de datos utilizada para dar soporte a la lógica del negocio, la parte transaccional y la explotación por parte del modelo de análisis predictivo, es generada por la aplicacion mediante Django REST Framerork, por defecto crea una base de datos en SQLite, la cual es una base de datos relacional ligera, pensada en dispositivos móviles o de bajos recursos, que no requiere de un servidor de base de datos y la cual se puede transportar fácilmente pues es contenida en un archivo, el framework se encarga de mapear el modelo, creando las tablas y relaciones pertinentes, a parte de las tablas propias del framework como las de configuración y las de autenticación de usuarios como se aprecia en la Figura **5-22**.

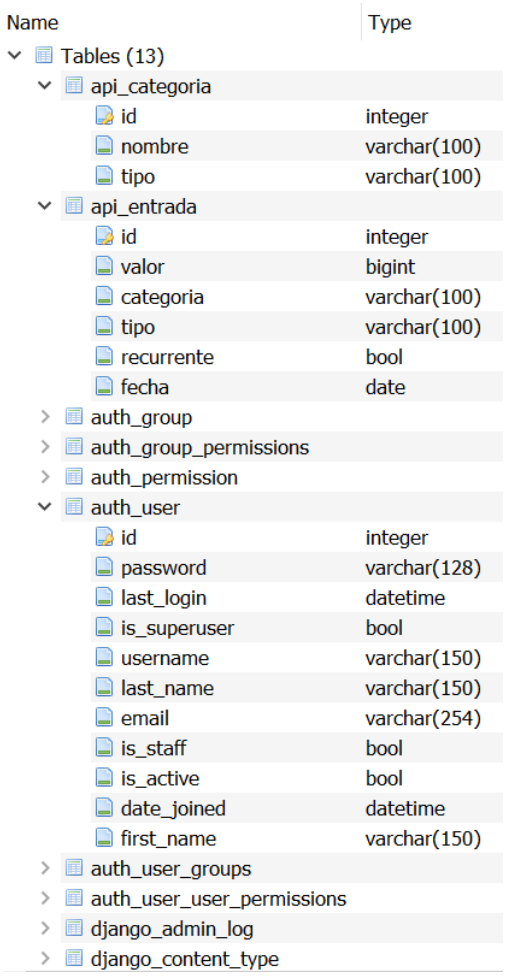


Figura 5-22.: Base de Datos - Estructura

A continuación, se analiza la estructura de la base de datos construida en torno a la lógica que da soporte al negocio de la aplicacion, la relación entre las principales entidades identificadas en el negocio, así como sus atributos, establecidos mediante autodeterminación, encontrando dos dependencias funcionales exclusivas DFE, como se aprecia en la Figura **5-23**, con base

en esta se diseña el diagrama notacional de conjuntos, identificando las relaciones entre las entidades, mediante las llaves foráneas por proceso FKP y las tablas adicionales por medio de las llaves foráneas por default FKD, representando en la Figura **5-24**.

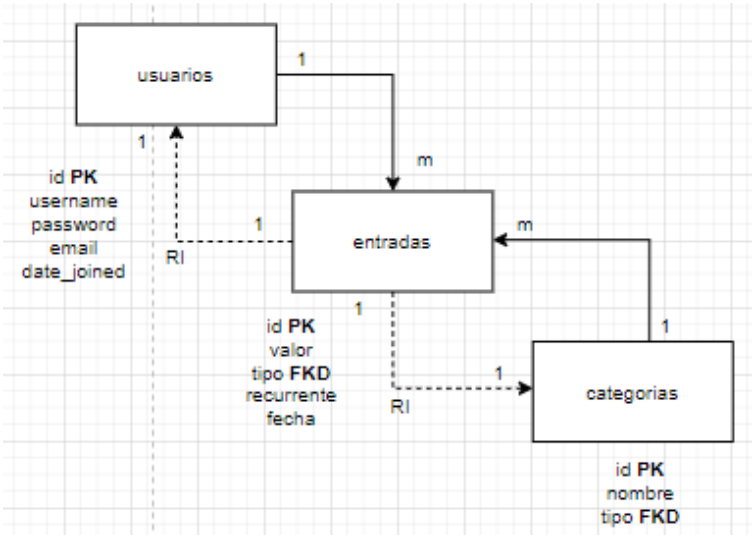


Figura 5-23.: Base de datos - Lógica negocio

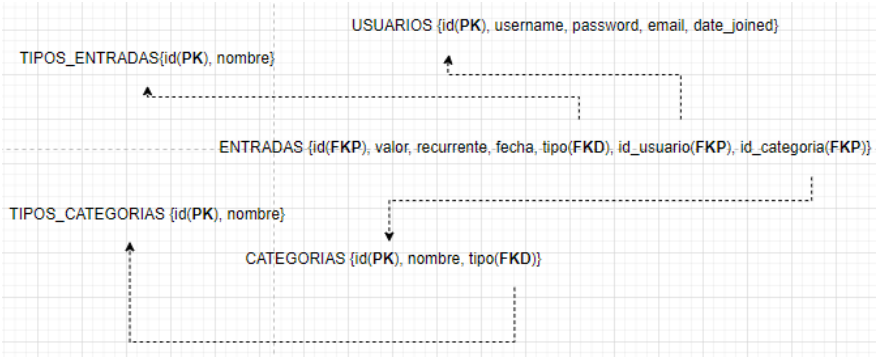


Figura 5-24.: Base de datos - Diagrama notacional de conjuntos

Con base en el diagrama notacional de conjuntos se procede a la creación del modelo conceptual (entidad-relación) Figura **5-25** y el modelo relacional Figura **5-26**, con base a estos diagramas se puede crear la base de datos mediante sentencias SQL, empezando a codificar del lado derecho del diagrama relacional al izquierdo, esto para respetar las relaciones de las entidades, las cuales garantizan su integridad referencial. En este caso el framework utilizado realiza dicha creación.

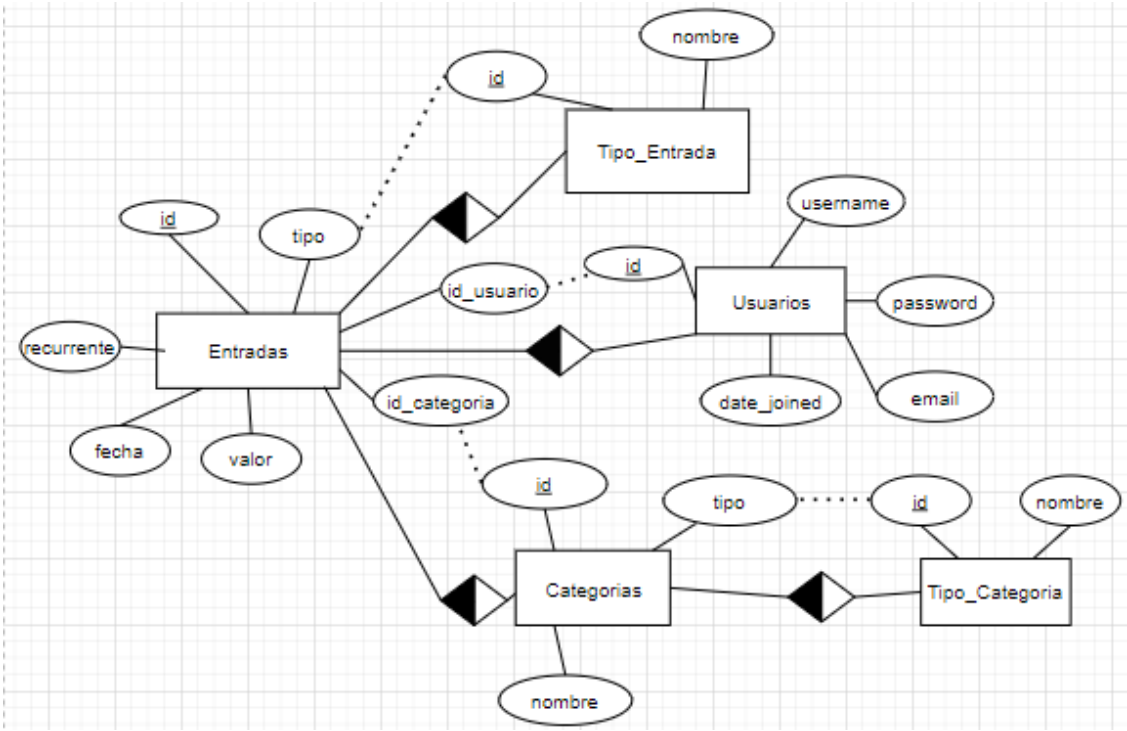


Figura 5-25.: Base de datos - Diagrama entidad-relación

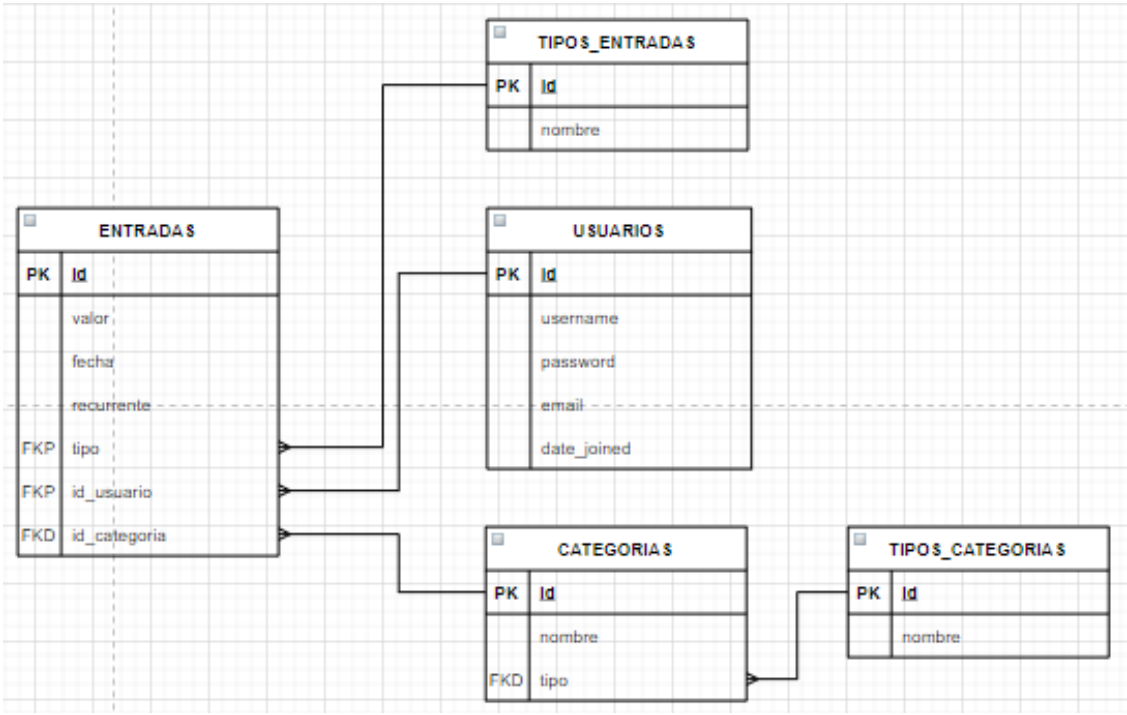


Figura 5-26.: Base de datos - Diagrama relacional

Para el posterior análisis descriptivo y predictivo mediante el lenguaje de programación Python, es necesario generar un dataset con la información obtenida de la base de datos, para este fin se agrupan los datos por mes de la base de datos producto de la interacción con la aplicacion, agrupando los gastos realizados por categoría al mes, el total de ingresos y gastos, esta información se exporta en un formato .csv para ser usado posteriormente en el modelo de análisis predictivo, la estructura de este dataset se puede observar en la Tabla **5-1**.

Ingresos	Categoria1	Categoria2	CategoriaN	Gastos	Saldo
mes1	mes1	mes1	mes1	mes1	mes1
mes2	mes2	mes2	mes2	mes2	mes2
mesN	mesN	mesN	mesN	mesN	mesN

Tabla 5-1.: Base de datos - Dataset para análisis

5.4. Modelo de análisis predictivo

Para realizar un análisis detallado de los gastos acumulados registrados en la aplicacion, se utilizo el lenguaje de programacion Python, mediante un Notebook en Jupyter, haciendo uso de los datos acumulados de la base de datos de la aplicacion, así como un dataset de entrenamiento con una gran cantidad de valores generados aleatoriamente siguiendo la estructura del archivo acumulado para poder entrenar un modelo de machine learning. El modelo elegido para el análisis predictivo fue el de arboles de decisión, después de realizar la investigación e implementación de los modelos mas usado, se eligió esta tecnica debido a que permite llevar a una clasificación de gastos por categorías de una manera grafica, siendo mas fácilmente identificable la pertinencia de realizar gastos en ciertas categorías, gracias a un conjunto histórico de datos, permitiendo realizar una mejor elección en cuanto a los gastos futuros, estableciendo una variable categórica parametrizada que nos permite saber si podemos llegar a un nivel de ahorro adecuado.A continuacion se detallara el proceso para realizar el análisis de los datos históricos de la aplicacion de finanzas personales.
Para la realización de la diferentes actividades de análisis se importan las siguientes librerías, y se lee el archivo de gastos acumulados obtenidos de la aplicación, el resultado se aprecia en la Figura **5-27**.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier

dsAnual = pd.read_csv("../dataset_anual.csv", sep=';')
dsAnual.head()
```

	Ingresos	Comida	Ropa	Transportes	Hogar	Educacion	Ocio	Prestamo	Otro	Gastos	Saldo
0	5358095	1237335	507286	145012	592868	344281	946495	925119	2562907	7261303	-1903208
1	6975917	1911081	351564	193776	183641	162882	465840	839995	818856	4927635	2048282
2	7311598	1090936	798628	33342	758919	301006	350264	344654	945257	4623006	2688592
3	7699153	656022	603521	98991	559431	332797	181382	591916	254798	3278858	4420295
4	9016808	952053	352031	4886	840671	344021	135504	720294	2058398	5407858	3608950

Figura 5-27.: DataSet Anual

Lo primero a la hora de hacer un análisis de datos es realizar una exploración de la data disponible, buscar datos faltantes o erróneos que nos puedan llevar a errores en el análisis, consultar los tipos de datos, realizar las respectivas conversiones, de variables numéricas o categóricas, según la necesidad de entradas o salidas del modelo de análisis que se piensa

crear, parte del proceso se puede observar en la Figura **5-28**.

```
dsAnual.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Ingresos     12 non-null     int64
 1   Comida       12 non-null     int64
 2   Ropa         12 non-null     int64
 3   Transportes  12 non-null     int64
 4   Hogar        12 non-null     int64
 5   Educacion    12 non-null     int64
 6   Ocio         12 non-null     int64
 7   Prestamo     12 non-null     int64
 8   Otro         12 non-null     int64
 9   Gastos       12 non-null     int64
 10  Saldo        12 non-null     int64
dtypes: int64(11)
memory usage: 1.2 KB
```

```
dsAnual.isnull().sum()
```

```
Ingresos       0
Comida         0
Ropa           0
Transportes    0
Hogar          0
Educacion      0
Ocio           0
Prestamo       0
Otro           0
Gastos         0
Saldo          0
dtype: int64
```

Figura 5-28.: DataSet Anual

Después de realizar el análisis previo de los datos y realizar los respectivos ajustes, se procede a ejecutar un análisis descriptivo de los datos, para esto se puede hacer uso de diferentes diagramas mediante los cuales se puede observar de una manera mas clara la composición y distribución de los datos, en Python se hace uso de librerías como matplotlib o seaborn, a continuacion se muestra el código en Python y las graficas construidas para el análisis descriptivo.

```
plt.plot(dsAnual['Ingresos'], label='Ingresos')
```

```
plt.plot(dsAnual['Gastos'], label='Gastos')
plt.xlabel('Mes')
plt.ylabel('Valor')
plt.title("Ingresos y Gastos Por Mes")
plt.legend()
```

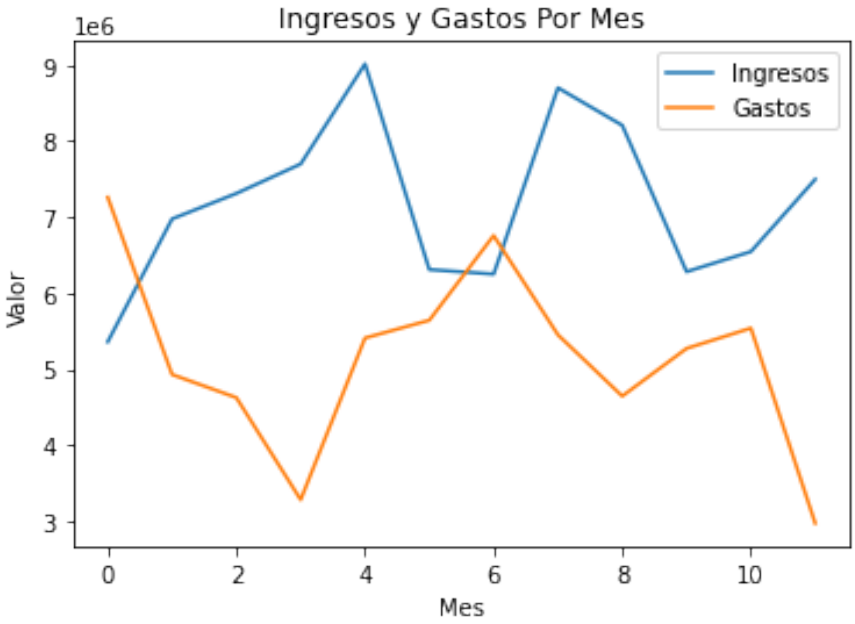


Figura 5-29.: Grafico ingresos y gastos por mes

```
columnas = dsAnual.columns.values.tolist()
columnasGastos = columnas[1:9]
valores = []

for i in columnasGastos:
valores.append(dsAnual[i].sum())

plt.pie(valores, labels = columnasGastos)
plt.title("Gasto anual por categorias")
plt.show()
```

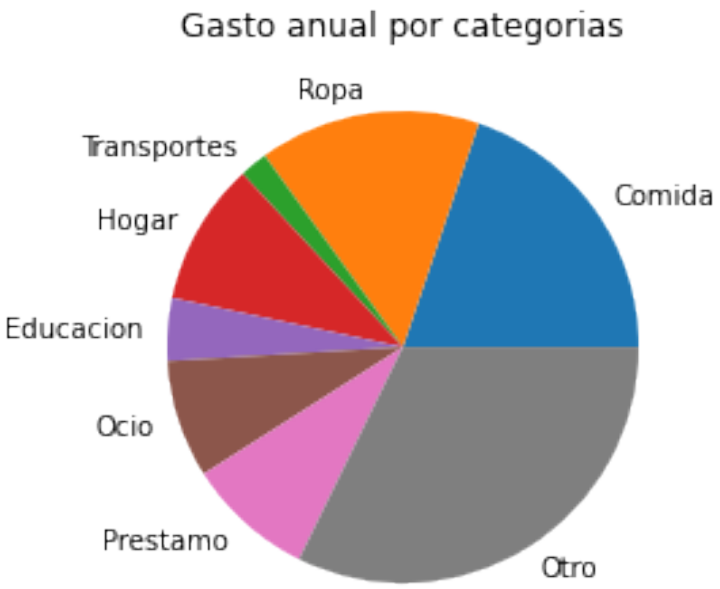


Figura 5-30.: Grafico gasto anual por categorías

```
plt.figure(1 , figsize = (20 , 10))
n = 0
for x in columnasGastos:
n += 1
plt.subplot(2 , 4 , n)
plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
sns.distplot(dsAnual[x] , bins = 20)
plt.title('Gastos en {}'.format(x))
plt.show()
```

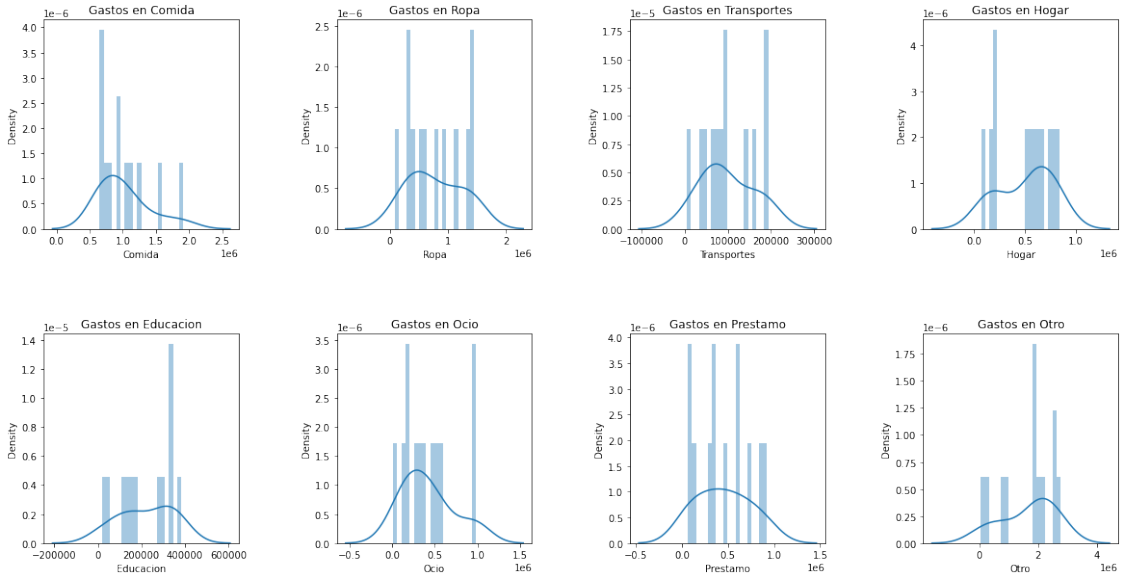


Figura 5-31.: Grafico distribución de gastos por cada categoría

Después de realizado el análisis descriptivo, se empieza a preparar y analizar un conjunto de datos de entrenamiento para le modelo predictivo, para esto se creo un conjunto de datos de gastos por categoría aleatorios, teniendo en cuenta un rango adecuado para cada variable según datos históricos almacenados en la aplicacion, en la exploración de los datos se observa que se cuenta con 300 entradas y 11 variables, ver Figura **5-32**, así como algunas graficas para el análisis visual de la data cargada, ver Figura **5-33**.

```
dsEntrenamiento = pd.read_csv("../dataset_entrenamiento.csv", sep=';')
dsEntrenamiento.head()
```

	Ingresos	Comida	Ropa	Transportes	Hogar	Educacion	Ocio	Prestamo	Otro	Gastos	Saldo
0	5779656	1235450	78062	75061	293441	80264	477998	205465	1753823	4199564	1580092
1	6752514	1783550	968224	64194	975258	586341	254035	118391	733193	5483186	1269328
2	9135616	1148578	1066450	89233	708111	365751	488151	786503	232525	4885302	4250314
3	6913781	1571950	797650	11302	855198	68925	501866	536899	245862	4589652	2324129
4	9279506	1726609	98274	29251	665702	366245	655526	867062	1296140	5704809	3574697

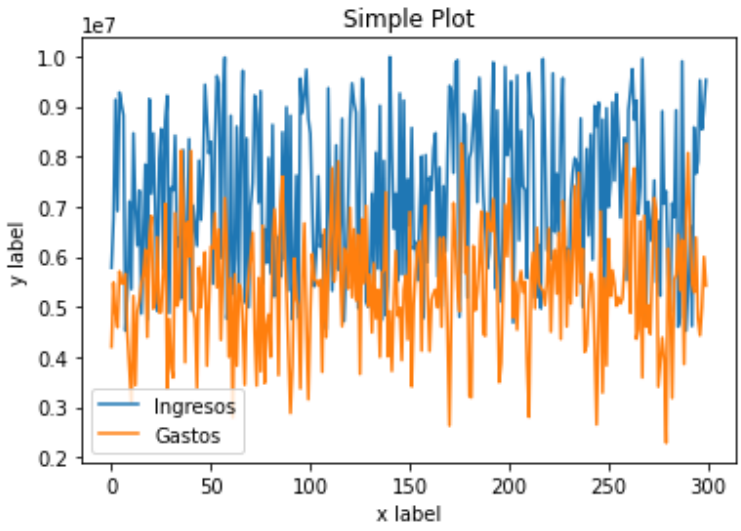
```
dsEntrenamiento.shape
```

```
(300, 11)
```

Figura 5-32.: Datos de entrenamiento

```
plt.plot(dsEntrenamiento['Ingresos'], label='Ingresos')
plt.plot(dsEntrenamiento['Gastos'], label='Gastos')
plt.xlabel('x label')
plt.ylabel('y label')
plt.title("Simple Plot")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x2dec87f70a0>
```




```
plt.hist(dsEntrenamiento.Gastos)
```

```
(array([ 6., 14., 26., 34., 51., 69., 43., 35., 13.,  9.]),
 array([2282862. , 2879728.7, 3476595.4, 4073462.1, 4670328.8, 5267195.5,
        5864062.2, 6460928.9, 7057795.6, 7654662.3, 8251529. ]),
 <BarContainer object of 10 artists>)
```

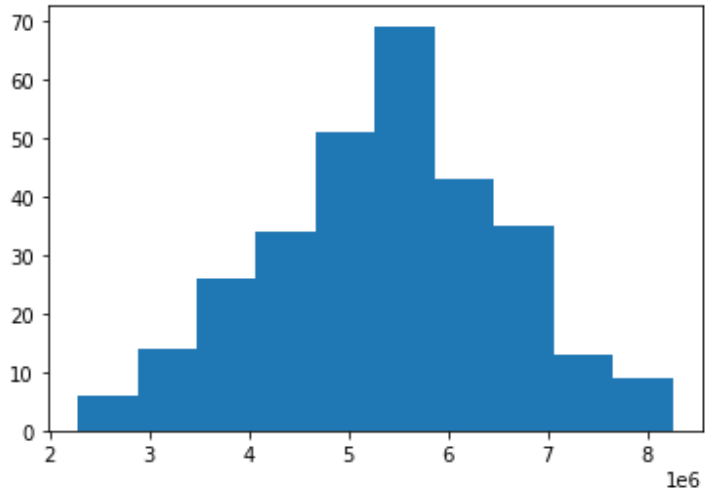


Figura 5-33.: Análisis grafico datos de entrenamiento

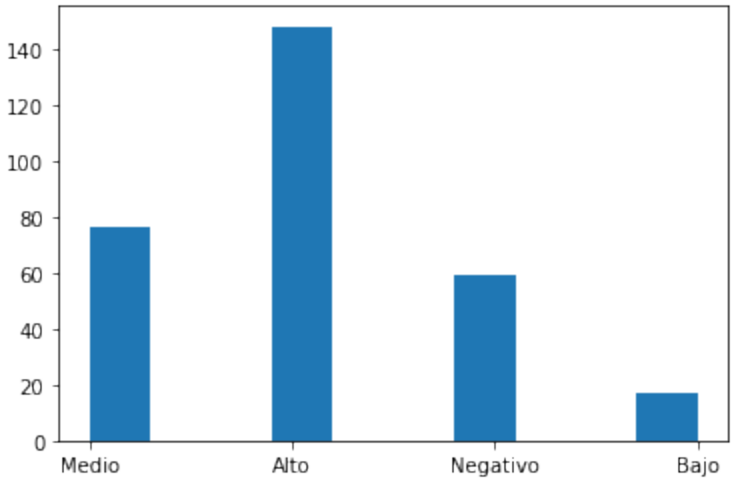


Figura 5-34.: Histograma variable ahorro

El objetivo de realizar el análisis predictivo mediante el árbol de decisiones es el de poder clasificar la pertinencia de gastos realizados en categorías claves, así como los montos adecuados para poder llegar a un nivel de ahorro, para esto se crea una variable categórica llamada ahorro con una reglas de saldos restantes del mes, que son el total de ingresos menos el total de gastos, esto mediante el siguiente código, y mediante el grafico de la Figura **5-34** se puede observar la cantidad la clasificación realizada en el conjunto de datos de entrenamiento.

```
condicionesAhorro = [
(dsEntrenamiento['Saldo'] < 0),
(dsEntrenamiento['Saldo'] > 0)
& (dsEntrenamiento['Saldo'] <= 500000),
(dsEntrenamiento['Saldo'] > 500000)
& (dsEntrenamiento['Saldo'] <= 2000000),
(dsEntrenamiento['Saldo'] > 2000000),
]

valores = ['Negativo', 'Bajo', 'Medio', 'Alto']

dsEntrenamiento['Ahorro'] = np.select(condicionesAhorro, valores)

plt.hist(dsEntrenamiento.Ahorro)
```

Lo siguiente es establecer las variables predictoras en esta caso las variables numéricas de gastos por categoría y la variable objetivo en este caso la categórica de ahorro. El conjunto de datos de entrenamiento se divide en un conjunto para entrenamiento y uno para prueba.

```
columnas = data.columns.values.tolist()
```

```
predictores = columnas[1:9]
objetivo = columnas[11]

data["es_entrenamiento"] = np.random.uniform(0,1, len(data))<=0.75
entrenamiento, prueba = data[data["es_entrenamiento"]==True], data[
```

Para la creación del árbol de decisiones se utiliza la librería sklearn en el siguiente código se aprecia la creación del árbol, este se alimenta con la data de entrenamiento y se calcula la predicción con la data de prueba y las variables predictoras.

```
tree = DecisionTreeClassifier(criterion="entropy",
min_samples_leaf=10, min_samples_split=30, random_state=0)
tree.fit(entrenamiento[predictores], entrenamiento[objetivo])

predicciones = tree.predict(prueba[predictores])
```

Se crea una tabla cruzada para ver la eficacia del entrenamiento, se compara la clasificación de los diferentes niveles de ahorro de la predicción con respecto a el entrenamiento, el resultado se aprecia en la Figura **5-35**.

```
#tabla cruzada para ver la eficacia del entrenamiento
pd.crosstab(prueba[objetivo], predicciones, rownames=["Actual"], colnames=["Predicciones"])
```

Predicciones	Alto	Medio	Negativo
Actual			
Alto	25	9	5
Bajo	3	2	2
Medio	12	4	3
Negativo	5	3	6

Figura 5-35.: Tabla cruzada entrenamiento

Para graficar el árbol de decisiones la librería exportgraphviz de sklearn, creando un archivo .dot, leyéndolo mediante graphviz, el resultado se aprecia en la Figura **5-36**, el criterio que se utilizo para la construcción del árbol fue el de entropía el cual se observa en cada nodo del mismo, mientras menos entropía mejor el resultado, podemos ir navegando por las ramas del árbol para ir mirando las variables predicadoras mas relevantes así como los valores de estos para llegar a clasificar dichos gastos en un nivel de ahorro determinado, cada nodo del árbol tiene una condicional en este caso las valores gastados en cada categoría para llegar a un nivel de ahorro, si se cumple la condición se va navegando a la izquierda, en caso contrario se navega a la derecha, se observa la cantidad de datos encontrados para cada condición y en la parte final del nodo se ve a que nivel corresponde o se aproxima mas estos valores, como valores de alto, bajo, medio o negativo.

```
from sklearn.tree import export_graphviz

with open("arbol_gastos.dot", "w") as dotfile:
export_graphviz(tree, out_file=dotfile, feature_names=predictores)
dotfile.close()

import os
os.environ["PATH"] += os.pathsep + 'C:/Program Files/Graphviz/bin'
from graphviz import Source

file = open("arbol_gastos.dot", "r")
text = file.read()
Source(text)
```

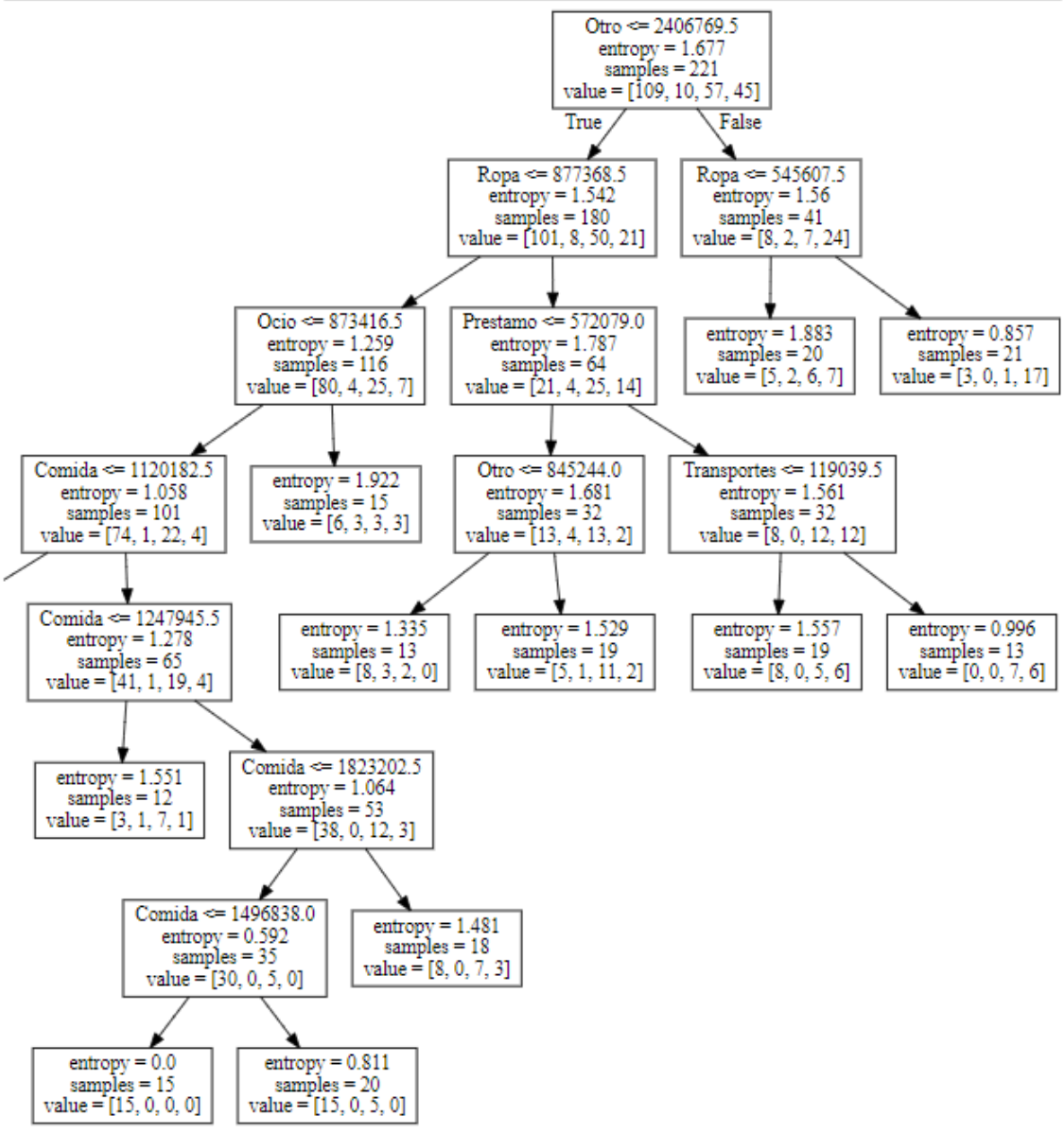


Figura 5-36.: Árbol de decisiones gastos personales

Parte III.

CIERRE DE LA INVESTIGACIÓN

6. RESULTADOS Y DISCUSIÓN

Con el desarrollo del presente proyecto, se logra presentar un prototipo de aplicación mediante el cual se puede solventar en alguna medida la problemática identificada y analizada de la falta de control de gastos personales. Como resultado de la investigación de las diferentes técnicas de análisis predictivo se elabora un modelo incorporado en la aplicación móvil para proveer un análisis descriptivo y predictivo de los ingresos y gastos ingresados en la aplicación.

Con el modelado de una arquitectura empresarial en el desarrollo del proyecto se establece una visión holística de una solución de software pensada en un contexto empresarial, mediante la incorporación de las diferentes capas definidas en la metodología ADM y TOGAF, se logra alinear la solución de software a los procesos y objetivos estratégicos del negocio de una organización.

La arquitectura de la aplicación orientada a servicios permite una alta escalabilidad, favoreciendo la reutilización y el desarrollo en paralelo de nuevas funcionalidades, la propuesta de una app móvil con una interfaz clara y sencilla, hace que el usar la aplicación se de de una forma natural en el publico objetivo.

7. CONCLUSIONES

7.1. Verificación, contraste y evaluación de los objetivos

A continuacion, se van a analizar los objetivos secundarios propuestos para el presente proyecto, como en el desarrollo del proyecto se fueron trabajando y como estos objetivos lleva al desarrollo del objetivo principal.

- Indagar acerca de las técnicas de machine learning mas usadas para el análisis predictivo, haciendo uso de las variables generadas por la aplicación móvil, proponiendo un modelo de análisis predictivo dentro de la aplicación para brindar un panorama claro del control de gastos personales.

Para el caso del primer objetivo especifico se realizo una investigación e implementación practica de los principales métodos de análisis predictivo, se evidencio que cada tecnica sirve a propósitos diferentes, seleccionado la tecnica de arboles de decisión para plantear un modelo de analisis predictivo.

- Crear una prototipo de aplicación haciendo uso del lenguaje de programación Python para implementar el modelo de análisis predictivo, creando un API para exponer servicios consumidos por una aplicación móvil, haciendo uso de tecnologías de desarrollo para aplicaciones móviles híbridas, brindando una aplicación de control y análisis de gastos en los dispositivos móviles Android y IOS.

Para este objetivo se plantea un prototipo de aplicacion móvil funcional, construido mediante una arquitectura orientada a servicios, con un componente de backend API REST construido con Python y Django Rest Framework y un componente de frontend construido en React Native.

- Diseñar una aplicación móvil intuitiva y fácil de usar mediante el desarrollo de una interfaz gráfica agradable, sencilla, con opciones fácilmente identificables para lograr que al usuario le sea claro su uso.

Este objetivo se consigue gracias al diseño de la interfaz de usuario desarrollado en la aplicación móvil, expuesto en el capitulo de Prototipo de aplicación.
La realización de estos objetivos en su conjunto llevan a la consecución del objetivo general del proyecto planteado:

"Diseñar un prototipo de aplicación móvil para el análisis y control de finanzas personales mediante la incorporación de análisis predictivo, haciendo uso de técnicas de machine learning que permita gestionar, analizar y controlar los gastos presentes y futuros".

7.2. Síntesis del modelo propuesto

7.3. Aportes originales

Como aporte original se presenta una aplicación móvil que cuenta con una arquitectura orientada a servicios con una interfaz intuitiva y fácil de usar, permitiendo llevar un control de los gastos personales por categorías, que utiliza los datos financieros (ingresos y gastos) para dar un panorama mas claro de los gastos mensuales apoyándose en graficas resultado de un análisis descriptivo. Se crea un modelo de analisis descriptivo y predictivo mediante el uso de la tecnica arboles de decisión, para poder predecir un nivel de ahorro según los gastos mensuales realizados en diferentes categorías.

7.4. Trabajos o publicaciones derivadas

Gracias a la arquitectura orientada a servicios de la aplicación, que brinda escalabilidad, se pueden seguir realizando desarrollos de componentes para complementar la aplicación de gestión de finanzas personales, por ejemplo un componente para planear y controlar ahorros programados, vinculación de pagos en linea, vinculación con entidades financieras para el control de créditos, modulo de reportes detallados con nuestro comportamiento financiero, entre otros. Otro trabajo derivado es el del mejoramiento del modelo de análisis predictivo en la aplicación.

8. PROSPECTIVA DEL TRABAJO DE GRADO

8.1. Líneas de investigación futuras

Como futuras lineas de investigación, se puede llegar a profundizar en la gestión de finanzas personales en Colombia, actualizar los datos mediante encuestas a nivel nacional para dicho fin, por otro lado se pueden evaluar diferentes técnicas de análisis predictivas que no fueron abordadas en este trabajo, especialmente técnicas de aprendizaje profundo como redes neuronales y ver el impacto en el análisis de finanzas personales y que usos prácticos pueden llegar a tener en este campo.

8.2. Trabajos de investigación futuros

Los trabajos de investigación futuros son los relacionados a las lineas de investigación descritas en la sección anterior, trabajos orientados al conocimiento del estado actual del control de finanzas personales por parte de la sociedad colombiana, así como trabajos relacionados al abordaje de técnicas de aprendizaje profundo para el analisis predictivo de las finanzas personales, haciendo uso del prototipo propuesto en este trabajo realizado.

Bibliografía

[1] J. et al. Aguilar. Estudio de pre-factibilidad para el diseño de una aplicación móvil para organizar y controlar las finanzas personales en colombia. Master's thesis, Corporación Universitaria Minuto de Dios, 2017.

[2] E. Arroyo. *Definición de principios de arquitectura para arquitectura empresarial de la organización*. Universidad Iberoamericana, 2015.

[3] P. Banerjee. Decision-tree classifier tutorial. https://www.kaggle.com/prashant111/decision-tree-classifier-tutorial. Accessed: 2021-05-19.

[4] bezkoder. Django: Post, put, get, delete requests example — rest apis. https://bezkoder.com/django-rest-api/. Accessed: 2021-05-13.

[5] S.J. Bolaños Castro, M. Ariza Riaño, and H.S. Alfonso Casallas. *Enfoque de Ingeniería de Software desde el proceso la arquitectura y la implementación*. Universidad Distrital Francisco José de Caldas, 2019.

[6] N. Braun, T. Hauth, C. Pulvermacher, and M. Ritter. An interactive and comprehensive working environment for high-energy physics software with python and jupyter notebooks. In *Journal of Physics: Conference Series*, volume 898, page 2. IOP Publishing, 2017.

[7] Sandra Milena Caicedo, Ligia Stella Bustos, and Jerónimo Rojas Diaz. Integración de procesos utilizando la arquitectura orientada a servicios-soa. *Scientia et technica*, 3(40):177, 2008.

[8] J. et al. Calvo. *Machine Learning, una pieza clave en la transformación de los modelos de negocio*. ManagementSolutions, 2018.

[9] M. Cristiá. Introducción a la arquitectura de software. Master's thesis, Universidad Nacional de Rosario, 2007.

[10] William Danielsson. React native application development. *Linköpings universitet, Swedia*, 10(4):10, 2016.

[11] Samuel Dauzon, Aidas Bendoraitis, and Arun Ravindran. *Django: Web Development with Python*. Packt Publishing Ltd, 2016.

[12] David GT Denison, Bani K Mallick, and Adrian FM Smith. A bayesian cart algorithm. *Biometrika*, 85(2):363–377, 1998.

[13] Elena Díaz García and Eduardo Cabrera Granado. Manual de uso de jupyter notebook para aplicaciones docentes, 2018.

[14] R. Dubey. React native architecture. https://www.educba.com/react-native-architecture/. Accessed: 2021-05-13.

[15] E Encina. Arquitectura orientada a servicios (soa) en la junta de andalucía. *Tecnimap Sevilla*, pages 1–6, 2008.

[16] F. Endel and H. Piringer. Data wrangling: Making data useful again. *IFAC-PapersOnLine*, 48(1), 2015.

[17] Facebook. React native. https://reactnative.dev/. Accessed: 2021-05-13.

[18] Michele Fratello and Roberto Tagliaferri. Decision trees and random forests. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, pages 374–383, 2018.

[19] W Garzon. Aplicación móvil para gestionar finanzas personales. Master's thesis, Universitaria Agustiniana Uniagustiniana, 2018.

[20] Alveiro Alonso Rosado Gomez and Juan Camilo Jaimes Fernández. Revisión de la incorporación de la arquitectura orientada a servicios en las organizaciones. *Revista Colombiana De Tecnologias De Avanzada*, 1(31):80, 2018.

[21] Joseph M Hilbe. *Logistic regression models*. CRC press, 2009.

[22] SToshl inc. Toshl finance mobil app. https://toshl.com/es/. Accessed: 2021-05-16.

[23] Jay Kreibich. *Using SQLite*. .º'Reilly Media, Inc.", 2010.

[24] Ashish Kumar. *Learning predictive analytics with Python*. Packt Publishing Ltd, 2016.

[25] A. Likas, N. Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2), 2003.

[26] M. López. Qué son las apps y tipos de apps. *Univirtual*, 2015.

[27] K. Mahindrakar. Customer segmentation (k-means). https://www.kaggle.com/kushal1996/customer-segmentation-k-means-analysis/notebook. Accessed: 2021-05-12.

[28] R Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Robert C. Martin Series(1st ed.)Pearson, 2017.

[29] Julián Andrés Mayor-Ríos, Diana Milena Pacheco-Ortiz, Juan Camilo Patiño-Vanegas, and Santiago Ernesto Ramos-y Yovera. Análisis de la integración del big data en los programas de contaduría pública en universidades acreditadas en colombia (analysis of the integration of big data into public accounting programs at accredited universities in colombia). *Revista CEA*, 5(9):57–58, 2019.

[30] Thomas W Miller. *Modeling techniques in predictive analytics with Python and R: A guide to data science*. FT Press, 2014.

[31] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[32] R. Reddy, M. Bruhn, and C. Tan. *Capacidades financieras en Colombia: resultados de la encuesta nacional sobre comportamientos, actitudes y conocimientos financieros*. Banco Internacional de Reconstruccion y Fomento / Banco Mundial, 2013.

[33] Jeanne W Ross, Peter Weill, and David Robertson. *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard business press, 2006.

[34] Sentry, Stream, ESG, Rollbar, Retool, and bit.io. Django rest framework. https://www.django-rest-framework.org/. Accessed: 2021-05-13.

[35] Sneha Shinde. Vehicle loan default prediction from Kaggle. https://www.kaggle.com/sneharshinde/ltfs-av-data. Accessed: 2021-05-12.

[36] O. Sotomayor. Logisic regressión loan card default exercise of Kaggle. https://www.kaggle.com/osmarsotomayor/logisic-regressi-n-loan-card-default/notebook?select=train.csv. Accessed: 2021-05-12.