



IMPACTO DE LA INGENIERIA DE LÍNEAS DE PRODUCTOS DE SOFTWARE EN LA  
CALIDAD DEL PROYECTO BASE Y LA EFICIENCIA DE SU DERIVACIÓN PARA  
APLICACIONES MÓVILES EN PRAGMA S.A.

DARRY ALEJANDRO MORALES JAIMES

Trabajo de grado

Asesor

Raúl Mazo Peña

UNIVERSIDAD EAFIT  
ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA  
MAESTRÍA EN INGENIERÍA  
MEDELLÍN  
2025

## CONTENIDO

pág.

0. RESUMEN .....	6
1. JUSTIFICACIÓN .....	7
2. ANTECEDENTES .....	9
3. PLANTEAMIENTO DEL PROBLEMA .....	13
4. OBJETIVOS .....	14
4.1. GENERAL .....	14
4.2. ESPECÍFICOS .....	14
5. DISEÑO METODOLÓGICO .....	15
5.1. ENFOQUE METODOLÓGICO .....	15
5.1.1. JUSTIFICACIÓN DEL ENFOQUE.....	15
5.1.2. TIPO DE INVESTIGACIÓN .....	15
5.2. DISEÑO DE LA INVESTIGACIÓN.....	16
5.2.1. DISEÑO CUASI-EXPERIMENTAL CON MEDICIONES PAREADAS .....	16
5.2.2. ESCENARIOS SELECCIONADOS (MUESTREO).....	16
5.2.3. VARIABLES DE ESTUDIO .....	17
5.2.3.1. VARIABLE INDEPENDIENTE: MÉTODO DE DERIVACIÓN .....	17
5.2.3.2. VARIABLE DEPENDIENTE 1: CALIDAD DEL PROYECTO BASE.....	18
5.2.3.3. VARIABLE DEPENDIENTE 2: EFICIENCIA DE DERIVACIÓN.....	19
5.2.3.4. VARIABLE DE CONTROL: CUMPLIMIENTO DE REQUISITOS MÍNIMOS.....	20
5.2.4. HIPÓTESIS DE LA INVESTIGACIÓN.....	20
5.2.5. ANÁLISIS ESTADÍSTICO .....	21
5.2.6. VALIDEZ DEL ESTUDIO .....	23
5.3. PROCEDIMIENTO EXPERIMENTAL.....	24
5.3.1. FASE 1: CARACTERIZACIÓN DEL ESTADO ACTUAL.....	24
5.3.2. FASE 2: ANALISIS DE DOMINIO Y DISEÑO DE LA LÍNEA DE PRODUCTOS DE SOFTWARE .....	25
5.3.3. FASE 3: IMPLEMENTACIÓN DEL CONFIGURADOR DE LÍNEA DE PRODUCTOS DE SOFTWARE .....	27
5.3.4. FASE 4: EXPERIMENTACIÓN CONTROLADA .....	36
5.4. CONSIDERACIONES ÉTICAS.....	41

6.	PRESUPUESTO.....	42
7.	RESULTADOS .....	43
8.	CONCLUSIONES.....	62
9.	RECOMENDACIONES.....	63
10.	REFERENCIAS.....	64
11.	ANEXOS .....	66
11.1.	EVALUACIÓN HEURÍSTICA .....	66
11.2.	CHECKLIST DE REQUISITOS MINIMOS (CRM) .....	69
11.3.	MOBILE APPLICATION ARCHITECTURE FEATURE MODEL – REDUCED OPERATIONAL (MAAFM-RO) .....	71
10.4	MOBILE APPLICATION ARCHITECTURE FEATURE MODEL (MAAFM) .....	74
10.5.	DOMAIN-SPECIFIC EXTENSIONS MODEL (DSEM).....	98

## LISTA DE TABLAS

pág.

Tabla I. TIEMPO DE DERIVACIÓN OBSERVADOS (TIEMPO EFECTIVO).....	43
Tabla II. DISTRIBUCIÓN DE TIEMPOS DE DERIVACIÓN POR COMPLEJIDAD DE ESCENARIO .....	44
Tabla III. DISTRIBUCIÓN DE TIEMPOS DE DERIVACIÓN POR TECNOLOGÍA .....	44
Tabla IV. CALIDAD DE PROYECTO BASE (IQ_ARQ) .....	45
Tabla V. DESAGREGACIÓN DE IQ_SONARCLOUD.....	47
Tabla VI. DESAGREGACIÓN DE IQ_HEURÍSTICA .....	48
Tabla VII. DISTRIBUCIÓN DE CALIDAD IQ_ARQ POR COMPLEJIDAD DE ESCENARIO .....	49
Tabla VIII. DIFERENCIAS PAREADAS OBSERVADAS EN EL EXPERIMENTO .....	50
Tabla IX. INTERVALOS DE CONFIANZA 95% (PARAMÉTRICO Y BOOTSTRAP).....	55
Tabla X. SÍNTESIS DE HALLAZGOS PRINCIPALES.....	59

## LISTA DE FIGURAS

pág.

Figura 1. Mobile Application Architecture Feature Model – Reduced Operational en VariaMos.....	27
Figura 2. Configuración de restricciones para MAAFM-RO en VariaMos .....	28
Figura 3. Configuración E1 en VariaMos.....	30
Figura 4. Configuración E2 en VariaMos.....	31
Figura 5. Configuración E3 en VariaMos.....	32
Figura 6. Configuración E4 en VariaMos.....	33
Figura 7. Configuración E5 en VariaMos.....	34
Figura 8. Configuración E6 en VariaMos.....	35
Figura 9. Tiempo de derivación por escenario y modalidad.....	45
Figura 10. Calidad del proyecto base derivado por escenario. ....	49
Figura 11. Comparativa de código entre derivación manual y derivación con SPL.....	57

## 0. RESUMEN

El mercado de aplicaciones móviles sigue experimentando un crecimiento acelerado y, en este contexto, la generación de proyectos base en organizaciones de desarrollo de software es vital y suele depender de procesos manuales con alta variabilidad en calidad y esfuerzo. Este trabajo evalúa el impacto de adoptar prácticas de ingeniería de líneas de productos de software (LPS) para derivar proyectos base de aplicaciones móviles en Pragma S.A. Bajo el paradigma Design Science Research, se diseñó un modelo formal de características para arquitectura de aplicaciones móviles (MAAFM) y su versión operacional reducida (MAAFM-RO) con 72 puntos de variabilidad; adicionalmente, se implementó un configurador en VariaMos e integraciones de generación semi-automática mediante Mason (Flutter), Gradle/CocoaPods y XCodegen (Android/iOS), logrando cubrir hasta un 80% de la variabilidad modelada para Flutter y 70% para tecnologías nativas. La propuesta se evaluó con un diseño cuasi-experimental pareado en seis escenarios (Flutter, Android e iOS para dominios financiero y retail), comparando derivación manual vs. derivación apoyada por LPS. La calidad se midió mediante el índice IQ\_ARQ, basado en ISO/IEC 25010 y compuesto por métricas de SonarCloud y evaluación heurística por ingenieros. Los resultados dejaron evidencia de una reducción promedio del tiempo de derivación del 49% y un incremento promedio de la calidad arquitectónica del 33%, con intervalos de confianza bootstrap consistentes. Con base en esto, se concluye que las LPS mejoran significativamente la eficiencia de derivación y la calidad del proyecto base, siendo sensible a la madurez y mantenimiento de los generadores y templates iniciales.

***Palabras clave:*** *Líneas de productos de software (LPS), modelo de características, variabilidad, arquitectura de aplicaciones móviles, derivación de proyectos base, calidad de software, VariaMos, mobile apps.*

## 1. JUSTIFICACIÓN

El mercado de aplicaciones móviles sigue experimentando un crecimiento acelerado, hasta el año 2023 en la principal tienda de aplicaciones, Google Play Store, existían 2.2 millones de aplicaciones disponibles para el público en general, siendo gratuitas el 98% de ellas [1]. Este tipo de activo digital requiere de especial atención en las decisiones arquitectónicas, componentes, tecnologías e integraciones seleccionadas dado que juegan un rol crítico en la eficiencia, escalabilidad y calidad del producto de cara a los usuarios [2].

En el contexto actual del desarrollo de software, especialmente en entornos empresariales de fábricas de software orientadas al desarrollo de dispositivos móviles, la capacidad para gestionar la variabilidad de sus productos de acuerdo con las características específicas de cada dominio de negocio es crucial [3]. Esta gestión permite garantizar la reutilización de componentes que derive en una reducción de tiempos y costos de desarrollo [4] y sirve para habilitar la posibilidad de iniciar nuevos desarrollos de software desde proyectos base con ciertas garantías de calidad.

En un estudio reciente en España [3], las aplicaciones móviles son el segundo tipo de producto digital más desarrollado por un grupo de 127 empresas dedicadas al desarrollo de software, de las cuales el 75% tiene experiencia mayor a 10 años en el mercado. De este grupo alrededor del 50% de las empresas solo identifica ocasionalmente las **características variables de sus productos**, y el 75% reporta que al comenzar un proyecto nuevo se analiza la **posibilidad de reutilizar componentes de productos existentes**. Esto es un claro indicativo que la adopción de prácticas industriales para la gestión de la variabilidad no está generalizada, especialmente entre las empresas desarrolladoras de aplicaciones móviles modernas.

En aras de favorecer los mecanismos de reutilización de componentes de código, promover la estandarización de los aspectos arquitectónicos (funcionales y no funcionales) y gestionar efectivamente la variabilidad de los productos de aplicaciones móviles, aún más compleja de gestionar en tiempos modernos [5], es necesario identificar el impacto de adoptar un proceso de producción orientado a la ingeniería de líneas de productos para gestionar la variabilidad de los activos de software de las empresas. En particular, con estas prácticas buscaremos que cuando evolucione la base (el *core*) de una línea de aplicaciones móviles, se pueda traducir en una reducción de los tiempos de configuración y disminución del esfuerzo inicial de desarrollo de nuevas aplicaciones. Además de este aspecto, estudiaremos las

garantías de escalabilidad, personalización y calidad ofrecidos por el nuevo proceso de producción, como aspectos fundamentales para mantener la competitividad en el mercado de las empresas de desarrollo de software.

## 2. ANTECEDENTES

En el ámbito de ingeniería de software, uno de los aspectos a considerar es la identificación y representación de las características comunes, dentro de un dominio particular, que permitan desarrollar componentes de software genéricos adaptables diseñados bajo un enfoque de reutilización [6]. Kyo Kang en diversas investigaciones realizadas para la universidad Carnegie Mellon en Estados Unidos, entre 1990 y 2002 [7], [8] propuso un método de análisis de dominio orientado a características (FODA) para (i) el descubrimiento sistemático de las similitudes entre diferentes aplicaciones de software que guardan relación entre sí y (ii) el modelado de las relaciones de comunalidad y variabilidad entre las diferentes características constitutivas de dichas aplicaciones. Este enfoque permite una generación de productos con una comprensión clara del dominio y de las características priorizadas por usuarios heterogéneos en el mercado [9].

Los lenguajes usados para modelar dichas características conforman una familia de lenguajes de ingeniería que sirven no solo para identificar las características comunes y variables, y las relaciones o restricciones entre dichas características, sino que sirve también como elemento de comunicación entre clientes e ingenieros de desarrollo [8]. Este mecanismo de comunicación es clave para entender la variabilidad de las características de un software según la perspectiva de los interesados [7]. En la historia de evolución de los lenguajes de modelado de características, FODA fue mejorado y de esas mejoras surgió FORM, un método de reutilización orientado a características en el cual varios conceptos, a la vez visuales tales como el uso de un círculo relleno en la parte superior de una característica determina que ésta es obligatoria, como semánticos, como el uso de las cardinalidades grupales para expresar las diferentes relaciones entre grupos de características sin las limitaciones de las relaciones de tipo “alternativa” de FODA o de las compuertas lógicas usadas por otros lenguajes de la misma familia [8].

Empresas como Nokia (Finlandia) y GoPhone (Estados Unidos), en 2002 y 2004, respectivamente, reconocieron la necesidad e implementaron líneas de productos para satisfacer su clientela de teléfonos móviles. Nokia [9] inició con una línea de productos de software (LPS) para la creación de navegadores genéricos que otros fabricantes y operadores de telecomunicaciones pudieran reutilizar, incluyendo grupos de pruebas como componentes reutilizables dentro de la LPS. Por su parte, GoPhone [10] desarrolló una LPS con una jerarquía de componentes reutilizables en Java 2 Micro Edition (J2ME) como capa de abstracción del hardware de sus dispositivos, que les permitió gestionar la variabilidad en sus sistemas de acuerdo con los requisitos de sus clientes, dado que dichos mecanismos habilitan la creación

(variante) de un activo base o fundamental por medio de la selección y modificaciones de las partes variables. [11]

En 2008, un caso de uso descrito por Jaejoon Lee [12] propone el uso de una LPS adaptada a sistemas con orientación a servicios para operar una posible oficina virtual del futuro (VOF). La propuesta incluía una arquitectura de componentes reusables de software que se integran con dispositivos físicos. En ese mismo año, en Corea del Sur, investigadores liderados por Jintae Kim [13] desarrollaron e implementaron exitosamente un marco de trabajo para realizar análisis de requerimientos de dominio y modelado de arquitectura (DRAMA), que permitió la adaptabilidad de las arquitecturas de dominio a las necesidades cambiantes del mercado en sistemas de e-learning, tarjetas de créditos y CRM, contemplando criterios de calidad y posibilidades de anticipación de cambios en los requerimientos.

Lee, para el año 2010, refinaría su propuesta de LPS para adaptarlo al desarrollo de activos reusables para sistemas basados en servicios [14], donde estos componentes ahora tienen una granularidad más adecuada para poder controlar las configuraciones de los servicios dinámicamente; nuevamente VOF fue su caso de prueba.

El desarrollo de aplicaciones móviles ha representado un reto importante desde la aparición de los *smartphone*, al tratarse de dispositivos heterogéneos en hardware y sistema operativo, inclusive dentro de una plataforma existen variaciones importantes de recursos del sistema [15], es por ello que en 2013 Rosa y Lucena [16] platearon AppSpotter como una infraestructura para componer dinámicamente esos artefactos de software, particularizados para cada dispositivo, de una forma automática mediante la caracterización de líneas de productos. Por su parte, Pavlič, logra desarrollar una aplicación para el manejo del estrés, @Life, reduciendo costos y mejorando calidad [17] mediante una línea de productos que gestiona la variabilidad de características entre versiones gratuitas, de pago y de pruebas, basándose en criterios de herencia y patrones de diseño.

Hacia el año 2015 se comienza a implementar arquitecturas flexibles para desarrollar aplicaciones para Android e iOS a partir de complementos o *plugins* [18]. En esas arquitecturas flexibles, los componentes implementados mediante *plugins* representan la variabilidad de las arquitecturas bases, homologando la comunicación entre componentes y plataformas por medio de interfaces de alto nivel. Por su parte, Gámez, propuso el enfoque FamiWare [19] como middleware para facilitar la creación de líneas de

productos de software dinámicas para aplicaciones móviles que se pueden auto adaptar al contexto de red, a la energía disponible y también a preferencias (por ejemplo, visuales) de los usuarios.

Entre 2016 y 2017 varias investigaciones realizan aportes importantes sobre la arquitectura de aplicaciones móviles para favorecer la reutilización de componentes. Bagheri planteó un principio arquitectónico basado en la invocación implícita por medio de mensajes asíncronos [20] mientras que Usman implementó MOPPET [21], una herramienta que genera automáticamente variantes de aplicaciones de acuerdo con su modelado de características, evidenciando una reducción del esfuerzo de desarrollo y mantenimiento. Por su parte Dürschmid, propuso diferentes estilos arquitectónicos aplicables a LPS para aplicaciones Android basado en su estudio de caso: “Los cinco estilos arquitectónicos propuestos son: (1) extensiones de actividad, (2) conectores de actividad, (3) entradas dinámicas de preferencias, (4) descripción de comportamiento desacoplada mediante archivos de configuración, y (5) un modelo de características utilizando recursos de Android” [22]. Sin embargo, pese a los diferentes avances y propuestas para la época, la implementación de la ingeniería de líneas de productos de software presentaba barreras importantes para su adopción, especialmente en organizaciones pequeñas donde podía existir resistencia cultural, conflictos por costos iniciales, y falta de visión clara de los productos a considerar [23].

Las diferencias entre plataformas Android e iOS también ha representado un impedimento para la concreción de LPS, y justamente este aspecto fue abordado por Dahlem en 2019 mediante su propuesta Smart Mobile [5]. Smart Mobile aborda conceptos de microservicios para garantizar portabilidad de módulos de funcionales que se integran a la pantalla principal de la app como características variables. Como evoluciones de esta propuesta tenemos que, por ejemplo, (i) Grua [24], propuso un modelo de características que analiza condiciones para la reconfiguración de la arquitectura de las apps en tiempo real de ejecución como mecanismo de autoadaptación, y (ii) Assunção prefirió el enfoque de generación automática de modelos de características para una línea de productos a partir de un intérprete de diagramas de clase UML en 2020 [25] para gestionar la variabilidad de las soluciones.

En años más recientes y en particular desde 2022 han surgido diversas aproximaciones para promover la implementación sostenible de las prácticas de ingeniería de LPS como la metodología para la evolución de las configuraciones de Nieke, que permitió actualizar líneas de productos de software hasta en un 55% de los casos estudiados [26]. La propuesta de Khoshnevis para generar líneas de productos basadas en

búsquedas que favorezcan la modularización y refactorización de su propio diseño [27] hacen de esta propuesta una opción más refinada de las ya conocidas SPL.

Los estudios hasta la fecha plantean la utilización de la ingeniería de líneas de productos para la generación automática de software como producto terminado más no como proyecto base inicial o arquetipo personalizado para cumplir con los requisitos de un dominio, bajo un enfoque tecnológico vigente en el de desarrollo de aplicaciones nativas y cross-platform.

### 3. PLANTEAMIENTO DEL PROBLEMA

Las investigaciones han demostrado la efectividad de la aplicación de SPL en desarrollos móviles, registrando aumentos de productividad por encima del 200% e impactos positivos en la calidad y costos de algunos tipos de aplicaciones como productos en 2021 [28]. Sin embargo, también es cierto que muchas organizaciones no son conscientes de las técnicas aplicables para este fin, dado que del 66% de las empresas que reconocieron productos base de dónde derivar nuevas aplicaciones, apenas entre el 20-25% de ellas tiene implementado prácticas ingenieriles y documentadas para el uso de líneas de productos de software que permitan generar productos eficazmente [4] [3]. Aunado a esto, la complejidad actual de desarrollo en términos de aplicaciones móviles radica en la diversidad de técnicas y tecnologías modernas para su construcción [2] por lo que cobra mayor relevancia el impacto que puede tener la ingeniería de líneas de productos para gestionar dicha variabilidad.

Pragma, S.A, una importante consultora de tecnología en Colombia, no es ajena a esta necesidad de caracterizar la gran variabilidad presente naturalmente en las soluciones móviles que la empresa desarrolla actualmente para sectores financieros, minoristas y aseguradores. Más aún si se contempla las vertientes de desarrollo nativo y cross-platform con sus respectivos patrones y tecnologías [2]. La complejidad que reviste un proceso de configuración de un proyecto o arquetipo base inicial, que reutilice componentes previamente desarrollados y contemple las buenas prácticas y estándares definidos hasta la fecha es alta, y demanda de un tiempo importante de perfiles técnicos especializados. Adicionalmente, dicho proceso manual no garantiza la completitud de decisiones trascendentales de arquitectura, necesarias para un producto de calidad consistente a lo largo del ciclo de desarrollo [29][30].

En este orden de ideas, la implementación de prácticas de ingeniería de LPS, aplicado al ámbito de desarrollo de aplicaciones móviles dentro de Pragma S.A, que (i) identifique y contemple la variabilidad tanto de sus líneas de negocio como las de sus líneas de desarrollo móvil importantes (nativo Android e iOS y cross-platform con Flutter) y (ii) valide configuraciones de proyectos base iniciales para cada producto particular, podría tener un impacto dentro de los procesos de la organización, los costos asociados a los tiempos de configuración inicial y la calidad del artefacto. Por lo tanto, la pregunta de investigación en la cual se enfoca este trabajo de maestría es la siguiente:

¿Cuál es el impacto de la ingeniería de líneas de productos de software en la calidad del proyecto base y la eficiencia de su derivación para aplicaciones móviles en Pragma S.A.?

## 4. OBJETIVOS

### 4.1.GENERAL

Evaluar el impacto de la ingeniería de líneas de productos de software en la calidad del proyecto base y la eficiencia de su derivación para aplicaciones móviles en Pragma S.A.

### 4.2.ESPECÍFICOS

- Objetivo Específico 1 (Obj-E1):  
Caracterizar el proceso actual de derivación de proyectos móviles base en Pragma S.A., estableciendo las métricas de línea base de calidad y eficiencia.
- Objetivo Específico 2 (Obj-E2):  
Identificar los puntos de variación y características comunes del proyecto base de aplicaciones móviles en Pragma S.A.
- Objetivo Específico 3 (Obj-E3):  
Diseñar un modelo de línea de productos de software (LPS) que represente la variabilidad identificada del proyecto base.
- Objetivo Específico 4 (Obj-E4):  
Implementar un mecanismo de derivación de proyectos base a partir del modelo de línea de productos de software diseñado.
- Objetivo Específico (Obj-E5):  
Comparar y evaluar el proceso actual de derivación versus el proceso basado en líneas de productos de software, midiendo el impacto en la calidad del proyecto base y la eficiencia de derivación.

## 5. DISEÑO METODOLÓGICO

### 5.1. ENFOQUE METODOLÓGICO

La presente investigación está planteada bajo el paradigma *Desing Science Research* (DSR) [31], el cual es apropiado para investigaciones que buscan diseñar, construir y evaluar artefactos de tecnología que intenten resolver un problema de tipo práctico bajo un contexto organizacional, como en este caso. Junto al paradigma DRS se aplicó un enfoque mixto [32], combinando métodos cuantitativos (medición de tiempo de derivación de producto y calidad de software) con métodos cualitativos (evaluación heurística y análisis de casos) para poder evaluar el proyecto base derivado tanto de forma manual como de forma sistemática, a través de la línea de productos de software propuesta para proyectos base de aplicaciones móviles en Pragma. S.A.

#### 5.1.1. JUSTIFICACIÓN DEL ENFOQUE

La aplicación del paradigma DRS se justifica, de acuerdo con Hevner [31] cuando la investigación cumple con las siguientes características:

- Diseño de un artefacto con base tecnológica. En este caso, el diseño de una solución de líneas de productos de software basado en un modelo de características para proyectos de aplicaciones móviles y su respectivo proceso de derivación mediante el software *VariaMos*.
- Resolución de un problema relevante, tal como el proceso de derivación para desarrollo *mobile* en Pragma S.A.
- Evaluación rigurosa del artefacto planteado como solución al problema, bajo un contexto real, como sucede con este estudio de caso organizacional.
- Existencia de una contribución al conocimiento científico, como el modelo de características que permitirá la derivación de proyecto base para soluciones móviles.

Esta investigación, obedeciendo al paradigma planteado, sigue 3 grandes etapas [33]: Diseño, desarrollo y evaluación del experimento.

#### 5.1.2. TIPO DE INVESTIGACIÓN

Esta investigación recurre al método aplicado de nivel explicativo que genera un conocimiento específicamente orientado a resolver un problema puntual dentro de la organización antes mencionada, explicando los elementos causales [34] por los cuales la implementación de prácticas de LPS impacta la calidad del proyecto base y la eficiencia de su derivación respecto a lo conseguido de forma manual actualmente.

## 5.2. DISEÑO DE LA INVESTIGACIÓN

### 5.2.1. DISEÑO CUASI-EXPERIMENTAL CON MEDICIONES PAREADAS

Se planteó y ejecutó un diseño cuasi experimental pareado, por cada escenario de derivación bajo el enfoque manual (grupo control) y semi-automatizado (grupo experimental) por la línea de productos de software, permitiendo aislar el efecto de la intervención [35].

El diseño pareado permite controlar la variabilidad entre los escenarios planteados y aporta un mayor poder estadístico que un diseño entre grupo, ya que cada escenario actúa como su propio control.

### 5.2.2. ESCENARIOS SELECCIONADOS (MUESTREO)

Los escenarios fueron seleccionados de acuerdo con el porcentaje de participación dentro del portafolio de la organización, bajo un muestreo intencional estratégico [36], quedando organizado de la siguiente manera:

ID	Tecnología	Dominio	Complejidad	Justificación
E1	Flutter	Financiamiento	Baja	Dominio de proyecto de mayor frecuencia, con aproximadamente 15% de participación en Pragma
E2	Flutter	Financiamiento	Alta	Mayor complejidad del tipo de proyecto más frecuente. 31% de participación en Pragma.
E3	Flutter	Retail	Baja	Segundo dominio más representativo, con participación aproximadamente de 10%
E4	Flutter	Retail	Alta	Mayor complejidad del segundo tipo de proyecto más frecuente. Equivalente a una participación del 23.97.5]Android

E6	iOS	Financiamiento	Medio	Validación portabilidad de línea de producto de software para iOS nativo con dominio financiero equivalente al 8%
----	-----	----------------	-------	---

El experimento se compuso de 6 escenarios totales bajo 2 modalidades de derivación, manual y semi automatizada por la línea de productos de software, para un total de 12 proyecto base analizados.

Formulación general del diseño del experimento:

- Escenario i (Ei) con (i=1..6)
- Especificación de requisitos común para cada escenario
- Derivación manual => Proyecto\_base\_manual\_i
  - Medición eficiencia\_derivación\_manual\_i
  - Medición calidad\_proyecto\_base\_manual\_i
- Derivación vía LPS => Proyecto\_base\_spl\_i
  - Medición eficiencia\_derivación\_spl\_i
  - Medición calidad\_proyecto\_base\_spl\_i
- Comparación pareada:
  - Delta\_eficiencia\_derivación =  

$$\text{eficiencia\_derivación\_spl\_i} - \text{eficiencia\_derivación\_manual\_i}$$
  - Delta\_calidad\_proyecto\_base =  

$$\text{calidad\_proyecto\_base\_spl\_i} - \text{calidad\_proyecto\_base\_manual\_i}$$

El tamaño de la muestra (N=6) puede limitar el poder estadístico para efectos pequeños, sin embargo, es apropiado para este tipo de estudio de casos múltiples en maestrías profesionales [37]. Adicionalmente se toma en cuenta que, al tratarse de caso de estudio de aplicación real, debe guardar correlación en tiempo a la ejecución de proyectos del portafolio de la organización durante la realización de esta investigación que se extendió hasta los 2 semestres.

### 5.2.3. VARIABLES DE ESTUDIO

#### 5.2.3.1. VARIABLE INDEPENDIENTE: MÉTODO DE DERIVACIÓN

Método de derivación: se entiende por método de derivación el proceso empleado para generar un proyecto base inicial que pueda ser empleado como punto de partida para un proyecto de desarrollo de

software de aplicaciones móviles en Pragma, a partir de una especificación mínima de requisitos arquitectónicos y funcionales.

#### **Niveles del método de derivación:**

- Derivación manual: se refiere al proceso tradicional basado en el uso de artefactos o arquetipos genéricos y que depende de la pericia de cada ingeniero de software para generar el proyecto base inicial desde cero, aplicando las mejores prácticas según su experiencia profesional.
- Derivación mediante LPS: se refiere al proceso propuesto de generación basado en la selección de características del modelo de variabilidad de la línea de productos de software desarrollado para semi-automatizar la creación de proyectos base inicial para proyectos de apps móviles.

#### **Operacionalización del método de derivación:**

- Derivación manual: el ingeniero a cargo inicia un proyecto base vacío desde el IDE de su preferencia y construye los requisitos mínimos requeridos de forma completamente manual.
- Derivación mediante LPS: el ingeniero a cargo configura las características mínimas requeridas en la herramienta de análisis de variabilidad seleccionada para este estudio (VariaMos), genera de forma automatiza el proyecto base inicial y depura o completa los aspectos específicos del contexto faltante de cada escenario, aquellos no soportados por el modelo de características propuesto.

#### 5.2.3.2. VARIABLE DEPENDIENTE 1: CALIDAD DEL PROYECTO BASE

Calidad del proyecto base (IQ-ARQ): se entiende como el grado de cumplimiento, que presenta el proyecto base derivado, respecto a los atributos de calidad deseables según interpretación de la norma ISO/IEC 25010:2911: mantenibilidad, fiabilidad, seguridad, eficiencia de desempeño y portabilidad.

#### **Operacionalización de IQ\_ARQ:**

$$IQ\_ARQ = ( 0.6 \times IQ\_SONARCLOUD ) + ( 0.4 \times IQ\_HEURISTICA )$$

$IQ\_SONARCLOUD$  = es un promedio ponderado y normalizado en una escala de (0-100) a partir de:

- Inverso ( Bugs/KLOC ) x 0.20
- Inverso ( Vulnerabilidades/ KLOC ) x 0.20
- Inverso (Code Smells/KLOC) x 0.15
- Inverso ( Promedio ( Complejidad ciclomática ) ) x 0.15

- Porcentaje de cobertura de código x 0.15
- Porcentaje de líneas de código duplicadas x 0.15

Donde KLOC = miles de líneas de código.

IQ\_HEURISTICA = es un promedio a partir de 2 ingenieros evaluadores independientes bajo escala Likert (1-5) convertida a (0-100). Los ingenieros valoran 4 atributos de calidad de la norma ISO 25010 [38]:

- Mantenibilidad: claridad arquitectónica y modularización.
- Fiabilidad: manejo de errores y validaciones de flujos alternos
- Seguridad: autenticación, encriptación y gestión de secretos.
- Portabilidad: independencia de plataformas y mecanismos de abstracción.

Los instrumentos usados para la medición de esta variable fueron:

- Reportes de sonar Cloud Enterprise
- Protocolo de evaluación heurística (Ver Anexo A).

### 5.2.3.3. VARIABLE DEPENDIENTE 2: EFICIENCIA DE DERIVACIÓN

Eficiencia de derivación (TIEMPO\_DERIVACIÓN): se define como el tiempo transcurrido (minutos) desde que inició el proceso de derivación hasta que se logró la construcción del proyecto base que cumplió los requisitos mínimos de completitud.

El criterio para detener la medición del tiempo se determina a partir del momento donde el proyecto base alcanza un nivel de cumplimiento de los requisitos mínimos del 100%, descrito en el punto siguiente de este informe.

#### **Operacionalización de TIEMPO\_DERIVACIÓN:**

$TIEMPO\_DERIVACIÓN = T\_FINAL - T\_INICIAL$  (min)

$T\_INICIAL$  = Timestamp del inicio del proceso

- Derivación manual:  $T\_INICIAL$  es la hora del primer commit del archivo readme vacío en el repositorio correspondiente.
- Derivación con LPS:  $T\_INICIAL$  es la hora de inicio del proceso de configuración en la herramienta VariaMos.

$T\_FINAL$  = Timestamp del momento cuando el proyecto logra el 100% de cumplimiento en el checklist de cumplimiento de requisitos mínimos (CRM). Ver Anexo B

Los instrumentos usados para la medición de esta variable fueron:

- Git hooks en los repositorios de código para determinación del T\_INICIAL a partir de un primer commit del archivo readme.md vacío.
- Los registros de tiempo de la plataforma de CI/CD para la validación de compilación, en este caso Azure Devops.
- Cronometraje manual para el proceso de ajustes finales que ocurrieron posterior al proceso de derivación.
- Checklist de cumplimiento de requisitos mínimos (CRM).

Es importante mencionar que el TIEMPO\_DERIVACIÓN para la derivación vía LPS incluye todas las fases de configuración en VariaMos, la generación semi-automática y los ajustes o correcciones manuales contextuales del dominio que se requieran. De esta forma se asegura la comparabilidad entre ambos procesos hasta el estado terminal cuando el proyecto base cumple con el CRM al 100%.

#### 5.2.3.4.VARIABLE DE CONTROL: CUMPLIMIENTO DE REQUISITOS MÍNIMOS

Cumplimiento de requisitos mínimos (CRM): se define como el grado de completitud, que presenta el proyecto base, con relación a los requisitos mínimos estructurales, funcionalidad y de calidad para poder considerarlo una derivación finalizada de forma exitosa.

##### **Operacionalización de CRM:**

$$\text{CRM} = (\text{Cantidad de item de checklist cumplidos} / 12) \times 100\%$$

CRM se utilizó como criterio de inclusión para poder realizar las comparaciones entre proyectos base que cumplieran con un CRM = 100%. Esto garantiza que las comparaciones se realizaron entre proyectos con estados de completitud equivalentes.

#### 5.2.4. HIPÓTESIS DE LA INVESTIGACIÓN

Este experimento plantea una expectativa teórica donde se espera que las prácticas de LPS mejoren la calidad del proyecto base derivado y reduzca el tiempo de derivación, por lo que se formularon 2 hipótesis de forma unilateral.

##### **H1: Calidad de proyecto del base**

$$H_0: \mu_{IQ\_ARQ\_SPL} \leq \mu_{IQ\_ARQ\_Manual}$$

$$H_1: \mu_{IQ\_ARQ\_SPL} > \mu_{IQ\_ARQ\_Manual}. (\text{con } \alpha=0.05)$$

“Los proyectos base derivados por vía de LPS presentan mayor calidad arquitectónica que los derivados manualmente”.

Esta hipótesis se plantea bajo la justificación teórica que los productos derivados por LPS ya incorporan buenas prácticas de arquitectura validadas previamente como parte de la generación semi-automatizadas a partir de una configuración válida del modelo de características, lo cual podría reducir la variabilidad atribuida al grado de experiencia o interpretación individual del ingeniero a cargo de la tarea.

## **H2: Eficiencia de derivación**

$H_0: \mu_{\text{TIEMPO\_DERIVACION\_SPL}} \geq \mu_{\text{TIEMPO\_DERIVACION\_Manual}}$

$H_1: \mu_{\text{TIEMPO\_DERIVACION\_SPL}} < \mu_{\text{TIEMPO\_DERIVACION\_Manual}}$ . (con  $\alpha=0.05$ )

“El proceso de derivación mediante LPS requiere un menor tiempo de derivación que el proceso manual para generar un proyecto base equivalente”.

Esta hipótesis se plantea bajo la justificación teórica que la automatización de tareas estructurales que son repetitivas (estructura de proyectos, inyección de dependencias, código boilerplate) reduce el tiempo de desarrollo, lo cual permite al ingeniero a cargo enfocarse en los aspectos específicos del dominio o contexto.

El nivel de significancia planteado  $\alpha=0.05$  se considera como convención estándar en ingeniería de software de forma empírica. No se consideró aplicar un proceso de corrección por comparaciones múltiples, dado que se realizaron 2 pruebas de hipótesis independientes, una por cada variable dependiente definida y se trata de un estudio de base exploratorio en un contexto organizacional industrial real, lo cual termina siendo congruente con las recomendaciones para estudios con muestra pequeñas donde el poder estadístico puede ser menor [39].

### 5.2.5. ANÁLISIS ESTADÍSTICO

El tamaño de la muestra de este estudio necesitó de un análisis estadístico cuantitativo robusto con un complemento cualitativo, que se ejecutó de la siguiente forma:

1. Se calcularon las diferencias pareadas  $\Delta_i = \text{Valor\_SPL}_i - \text{Valor\_Manual}_i$  para cada variable independiente: IQ\_ARQ y TIEMPO\_DERIVACIÓN.
2. Se evaluó la normalidad de las diferencias para determinar la aplicación de Prueba T pareada unilateral si se cumplía con el criterio de distribución normal o Prueba Wilcoxon si no se prestaba una distribución normal. La normalidad se terminó a partir del valor de  $p \geq 0.05$  para la asunción de distribución normal, caso contrario sería una distribución no normal.
  - a. Caso de distribución normal:
    - i. Prueba T de Student pareada unilateral

ii. Estadístico de prueba:  $t = (\bar{D} - 0) / (S_d / \sqrt{n})$

iii. Donde:

$\bar{D}$  = media de diferencias pareadas

$S_d$  = desviación estándar de diferencias

$n = 6$  pares

Grados de libertad:  $df = n - 1 = 5$

iv. Región de rechazo:

$H_1: \mu_{SPL} > \mu_{Manual} \rightarrow$  Rechazar  $H_0$  si  $t > t_{(\alpha,df)}$

$H_1: \mu_{SPL} < \mu_{Manual} \rightarrow$  Rechazar  $H_0$  si  $t < -t_{(\alpha,df)}$

b. Caso de no distribución normal:

i. Prueba de Wilcoxon de rangos con signo (unilateral)

ii. Estadístico de prueba:  $W$  (la suma de rangos positivos)

iii. Región de rechazo según tabla de valores críticos para  $n=6$

3. Se evaluó el tamaño del efecto para determinar su relevancia práctica:

a. Prueba de Cohen's  $d$  pareado:  $d = \bar{D} / S_d$

i.  $|d| < 0.5 \rightarrow$  Implica un efecto pequeño

ii.  $0.5 \leq |d| < 0.8 \rightarrow$  Implica un efecto mediano

iii.  $|d| \geq 0.8 \rightarrow$  Implica un efecto grande

iv.  $|d| \geq 1.3 \rightarrow$  Implica un efecto muy grande

4. Se generó una prueba de intervalos de confianza o (Bootstrapping) con 10000 muestras, dado la potencia de la muestra  $n=6$  [40].

5. Se aplicó un análisis cualitativo a partir de:

a. Análisis de casos individuales

i. Comparación de proyectos base derivados manual o con LPS con estructuras de código frente a frente.

ii. Identificación de las diferencias arquitectónicas significativas.

iii. Análisis de las métricas procesadas por SonarCloud de forma desagregada.

b. Análisis cruzado de casos

i. Valoración de las características que, de forma consistente, generaron mayor diferencia de tiempos.

- ii. Determinación de los aspectos arquitectónicos donde la LPS propuesto generó mayor beneficio.

#### 5.2.6. VALIDEZ DEL ESTUDIO

##### **Validez interna**

- Amenaza: Algún evento externo afecte los resultados del experimento.  
Mecanismo de control: Los proyectos base se derivaron durante un lapso determinado (1,5 meses) y bajo las mismas condiciones de quipos e infraestructura.
- Amenaza: Los ingenieros a cargo de los procesos de derivación aprenden y tienen un mejor desempeño a lo largo del tiempo.  
Mecanismo de control: Un mismo ingeniero fue seleccionado para ejecutar cada par de derivaciones (manual y vía LPS) de un escenario particular.
- Amenaza: Posibilidad de criterios alternos en la fase de experimentación.  
Mecanismo de control: se generaron los formatos claros y específicos para la evaluación (ver anexos), junto al protocolo de evaluación heurística.
- Amenaza: Los evaluadores pueden influenciar los resultados según sus criterios y expectativas.  
Mecanismo de control: Los evaluadores a cargo de la heurística no conocen el método de derivación usado para obtener el proyecto base evaluado.
- Amenaza: Riesgo de no determinar un efecto real dado el poder estadístico de la muestra  $N=6$ .  
Mecanismo de control: Complementar las pruebas usando tamaños de efecto Cohen's  $d$  y *bootstrapping*.
- Amenaza: Resultados con distribuciones no normales pueden invalidar resultados de prueba  $t$ .  
Mecanismo de control: Se realiza verificación de distribución normal con prueba Shapiro-Wilk, en caso negativo se usa prueba Wilcoxon.

##### **Validez externa:**

Se reconocen algunas de las limitaciones en las generalizaciones de este estudio:

- El estudio se centra en estudio de caso para la empresa Pragma, S.A. y su contexto regional y dominios de negocio, por lo que los resultados no podrán extrapolarse naturalmente a otras organizaciones.

- La validación empírica se enfoca en tecnologías Flutter, Android e iOS nativo, por lo que los resultados no podrán extrapolarse naturalmente a otras tecnologías sin las respectivas validaciones del caso.
- Se consideran únicamente los dominios de negocio de app financieras y *retail*, por lo que la validez de estos resultados a otros dominios de negocio se debe abordar en experimentos futuros.
- Los procesos de derivación para obtención de proyecto base fueron ejecutados por ingenieros de software con un rango de experiencia entre 3-4 años por tecnología objetivo, por lo que los resultados pueden variar con otros perfiles tales como *senior*.

### 5.3. PROCEDIMIENTO EXPERIMENTAL

#### 5.3.1. FASE 1: CARACTERIZACIÓN DEL ESTADO ACTUAL

Objetivo específico asociado: Caracterizar el proceso actual de derivación de proyectos móviles base en Pragma S.A, estableciendo la línea base de las métricas relacionadas con calidad y eficiencia.

Actividades (8 semanas):

1. Se seleccionaron 6 proyectos representativos de aplicaciones móviles que se hayan ejecutados entre el lapso de septiembre 2024 y Julio 2025 para su respectiva revisión.
  - a. Los criterios de inclusión fueron:
    - i. Proyectos en su fase inicial con una cantidad de commits menor a 200.
    - ii. Proyectos bajo tecnologías Flutter, Android o iOS.
    - iii. Proyectos bajo dominios de negocio: Financial y Retail.
    - iv. Proyectos con disponibilidad de acceso a repositorios de código y sonar cloud.
2. Proyectos que presenten un complejidad baja o complejidad alta. Se determinó de cada proyecto el primer *commit* con proceso de CI/CD exitoso (*checklist* CRM) para obtener el proyecto base inicial en cada proyecto.
3. Se determinó el IQ\_SONARQUBE del proyecto base inicial ejecutando un análisis de su código fuente bajo SonarCloud bajo la configuración predeterminada de la herramienta sobre los lenguajes Dart, Kotlin y Swift.
4. Se estimó en forma retrospectiva los tiempos (minutos) consumidos en la derivación manual del proyecto base inicial previamente analizados. Para ello, se consideraron 2 casos:
  - a. Proyectos que presentaron múltiples *commits*, se calculó el tiempo transcurrido entre el *commit* 0 y el primer *commit* con *checklist* CRM exitoso.

- b. Proyectos con un *commit* único, pero con cumplimiento de *checklist* CRM fueron estimados a partir de entrevistas estructuradas con los ingenieros responsables basado en Jørgensen [42], buscando el consenso entre dichos ingenieros.

### 5.3.2. FASE 2: ANALISIS DE DOMINIO Y DISEÑO DE LA LÍNEA DE PRODUCTOS DE SOFTWARE

Objetivo específico asociado:

- Identificar los puntos de variación y características comunes del proyecto base de aplicaciones móviles en Pragma S.A.
- Diseñar un modelo de línea de productos de software que represente la variabilidad identificada del proyecto base.

Actividades (4 semanas):

- Se aplicó la técnica FODA (Feature-Oriented Domain Analysis [43]) de acuerdo con los proyectos previamente identificados como fuente de información. Adicionalmente se contó con:
  - Repositorios de arquetipos de arquitectura para aplicaciones móviles para las tecnologías implicadas en este estudio: Flutter, Android e iOS.
  - Resultados de 4 entrevistas con stakeholders importantes:
    - 1 arquitecto de software aplicaciones móviles.
    - 2 ingenieros de seniority semi-senior con especialidad en Flutter y desarrollo nativo Android/iOS respectivamente.
    - 1 ingeniero DevSecOps.
  - Se ejecutó una sesión bajo la técnica de brainstorming para identificar los aspectos comunes entre proyectos que fueran la base de variabilidad.
  - Se construyó una taxonomía preliminar con las categorías de principales de la variabilidad.
  - Se validó dicha taxonomía con los stakeholders entrevistados previamente.
- Se construyó el modelo de características en formato markdown (md) con todas las categorías, subcategorías, cardinalidades, y restricciones que aplicarían. (Mobile Application Architecture Feature Model (MAAFM))
- Se validó que las características planteadas hayan sido contempladas por los proyectos base derivados manualmente seleccionados en fases previas.

- Se derivó un modelo de características operacional reducido (MAAFM-RO – Ver Anexo C), con el objetivo de poder garantizar la posibilidad de implementación, durante la experimentación, dentro del sistema VariaMos, tomando en cuenta el proceso de generación semi-automática planteada.
  - Se identificaron las 72 características relevantes en cualquier proceso de derivación.
  - Se filtraron características implementables bajo generadores semi-automatizados: Mason para flutter y Gradle/CocoaPods script para nativo Android/iOS
  - Se validó la cobertura del modelo reducido con respecto a las características presentes en los proyectos base previamente seleccionados.

Contribución académica: Se propuso como parte de este estudio un modelo formal de características para arquitectura de aplicaciones móviles (MAAFM) que puede ser empleado en la industria para derivaciones de proyectos base, arquetipos o productos a partir de una línea de productos de software particular. Dicho modelo propone también características acordes a diversos dominios o campos de aplicación.

Suficiencia del modelo operacional reducido: la reducción del modelo de características hacia el modelo operacional reducido (MAAFM-RO) para la experimentación empírica obedece a criterios metodológicos reconocidos en estudios de adopción de líneas de productos de software. Restringir el modelo experimental a aquellas características que varían verdaderamente entre las configuraciones evaluadas y cuya implementación pueden ser verificable de manera objetiva permite maximizar el control sobre el experimento y reducir el sesgo en el mismo. [44][45].

Criterios usados para la construcción del modelo operacional reducido:

- Discriminación experimental: las características seleccionadas permitieron diferenciar tecnológica, arquitectónica y funcionalmente los escenarios de derivación planteados, de tal forma que se cubrió una variabilidad relevante dentro del experimento.[46]
- Restricciones y dependencias: el modelo operacional reducido mantuvo una cobertura de las restricciones cardinales del modelo original y las restricciones lógicas que fueron críticas para la realidad de los escenarios derivados, lo cual permitió mantener el foco en la validación de la línea de productos de software. [45]
- Verificabilidad: la reducción del modelo permitió establecer listas de verificación claras sobre las implementaciones de las características de forma objetiva, lo que aseguró la consistencia del

experimento y la capacidad de reproducirlo a futuro, evitando un posible “modelo inflado” que pudo afectar la validez de las pruebas. [47].

### 5.3.3. FASE 3: IMPLEMENTACIÓN DEL CONFIGURADOR DE LÍNEA DE PRODUCTOS DE SOFTWARE

Objetivo específico asociado: Implementar un mecanismo de derivación de proyectos base a partir del modelo de línea de productos de software diseñado.

Actividades (2 semanas):

- Se generó la configuración del modelo de características operacional reducido (MAAFM-RO) dentro de la plataforma VariaMos (<http://variamos.com/>) bajo el lenguaje *Feature model without attributes*.
  - Se importaron las 72 características del modelo operacional con sus jerarquías y cardinalidades.

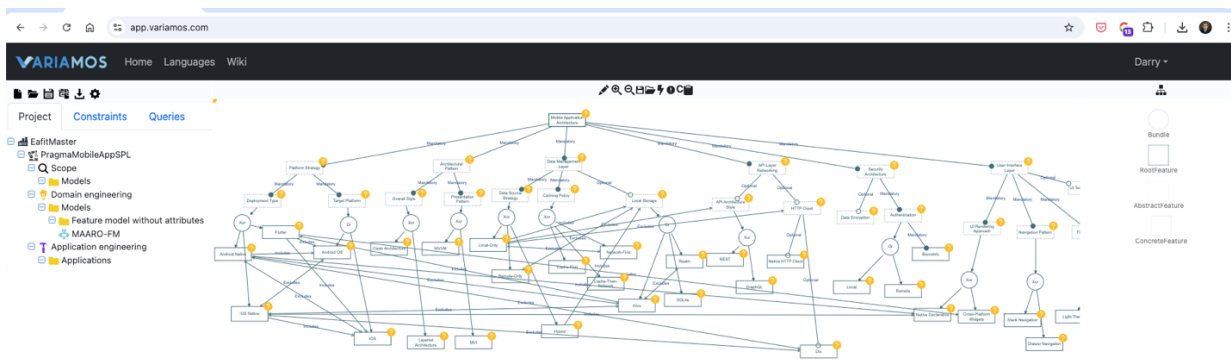


Figura 1. Mobile Application Architecture Feature Model – Reduced Operational en VariaMos

- Se configuraron las restricciones entre características que no pudieron ser representadas en el modelo de características, utilizando el lenguaje CLIF (*Common Logic Interchange Format*):
  - ( $\Rightarrow$  Flutter (and Android iOS))
  - ( $\Rightarrow$  AndroidNative Android)
  - ( $\Rightarrow$  iOSNative iOS)
  - ( $\Rightarrow$  Flutter (and CrossPlatformWidgets Dio))
  - ( $\Rightarrow$  iOSNative (and NativeDeclarative NativeHTTPClient))
  - ( $\Rightarrow$  PushNotifications PushProvider)
  - ( $\Rightarrow$  PushNotifications)

- (or FirebaseCloudMessaging AirShip OneSignal))
- (not (and FirebaseCloudMessaging AirShip))
- (not (and FirebaseCloudMessaging OneSignal))
- (not (and AirShip OneSignal))
- (=> LocalOnly (and (not REST) (not GraphQL) (not NativeHTTPClient) (not Dio)))
- (=> (and REST (or AndroidNative iOSNative)) NativeHTTPClient)
- (=> (and GraphQL Flutter) Dio)
- (=> (and GraphQL (or AndroidNative iOSNative)) NativeHTTPClient)
- (=> Hive Flutter)
- (=> (not Flutter) (not Hive))
- (=> SQLite (or AndroidNative iOSNative))
- (=> MVI CleanArchitecture)
- (=> (and Financial Alta) (and Biometric DataEncryption))

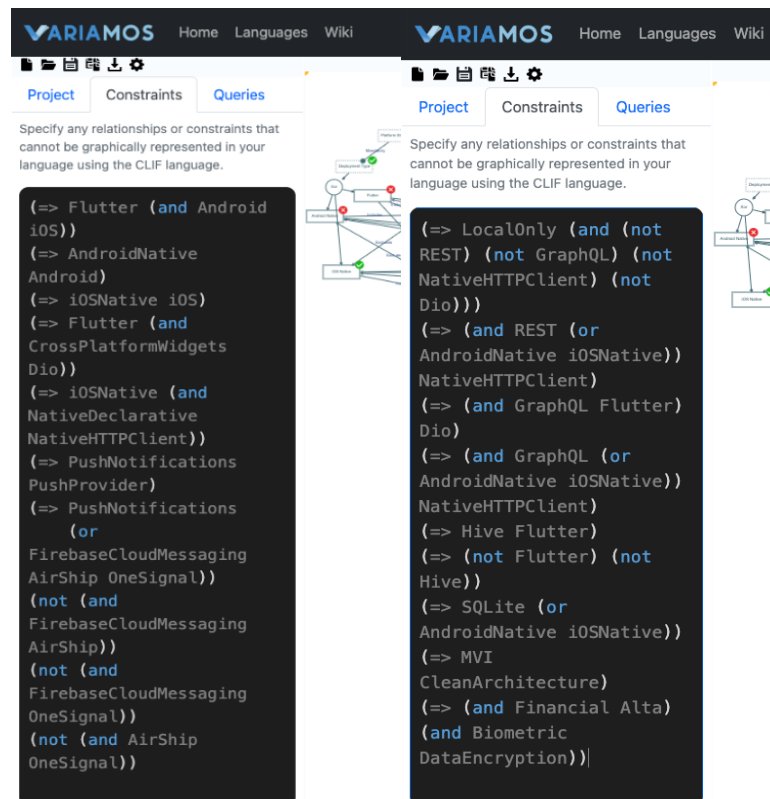


Figura 2. Configuración de restricciones para MAAFM-RO en VariaMos

- Se configuraron los 6 escenarios seleccionados para la ejecución de la derivación semiautomática del experimento, posteriormente se ejecutaron pruebas de pre-validación con configuraciones válidas e inválidas para comprobar el funcionamiento del modelo de características:
  - E1 - Aplicación Flutter para dominio financiero de baja complejidad:
    - Deployment Type: Flutter
    - Target Platform: Android = TRUE, iOS = TRUE
    - Overall Style: Layered Architecture
    - Presentation Pattern: MVVM
    - Data Source Strategy: Remote-Only
    - Local Storage: Hive
    - Caching Policy: Network-First
    - API Architecture Style: REST
    - HTTP Client: Dio
    - Authentication: Remote
    - Biometric: FALSE
    - Data Encryption: TRUE
    - UI Rendering: Cross-Platform Widgets
    - Navigation: Stack
    - Theme Support: Light Theme
    - UI Testing: FALSE
    - Camera: FALSE
    - Location Services: FALSE
    - Push Notifications: FALSE
    - Domain: Financial
    - Account Management: TRUE
    - Transactions: TRUE
    - Payment Methods: FALSE

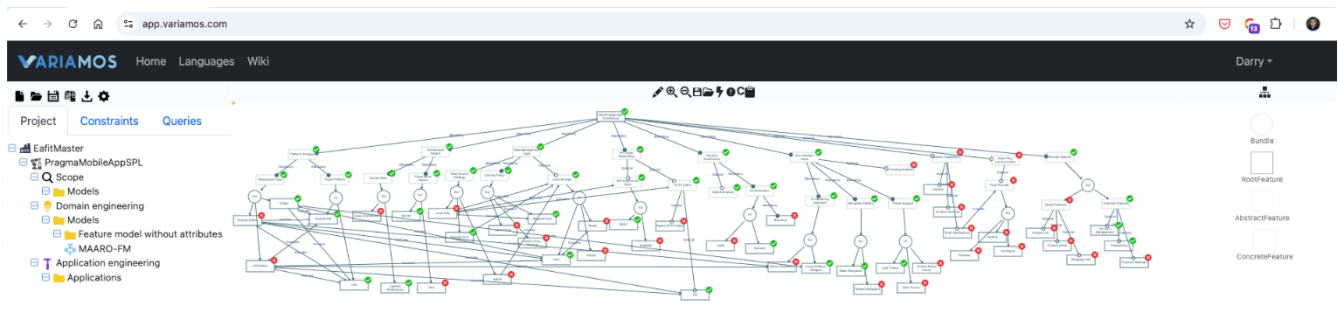


Figura 3. Configuración E1 en VariaMos

- E2: Aplicación Flutter para dominio financiero de alta complejidad:
  - Deployment Type: Flutter
  - Target Platform: Android = TRUE, iOS = TRUE
  - Overall Style: Clean Architecture
  - Presentation Pattern: MVI
  - Data Source Strategy: Hybrid
  - Local Storage: Hive # SQLite prohibido en Flutter
  - Caching Policy: Cache-Then-Network
  - API Architecture Style: GraphQL
  - HTTP Client: Dio
  - Authentication: Remote
  - Biometric: TRUE
  - Data Encryption: TRUE
  - UI Rendering: Cross-Platform Widgets
  - Navigation: Drawer
  - Theme Support: Light Theme, Dark Theme, Custom Brand Colors
  - UI Testing: TRUE
  - Camera: TRUE
  - Location Services: TRUE
  - Push Notifications: TRUE
  - Push Provider: Firebase Cloud Messaging
  - Domain: Financial
  - Account Management: TRUE
  - Transactions: TRUE

- Payment Methods: TRUE

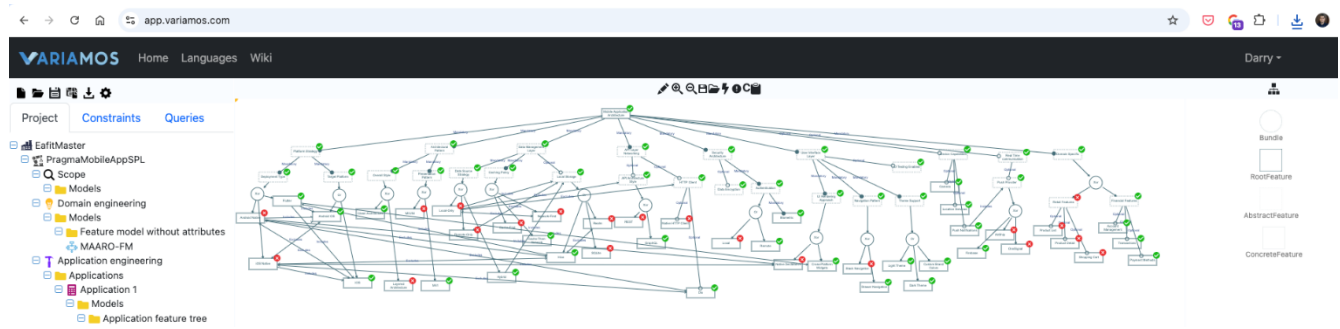


Figura 4. Configuración E2 en VariaMos

▪ E3: Aplicación Flutter para dominio *retail* de baja complejidad:

- Deployment Type: Flutter
- Target Platform: Android = TRUE, iOS = TRUE
- Overall Style: Layered Architecture
- Presentation Pattern: MVVM
- Data Source Strategy: Remote-Only
- Local Storage: Hive
- Caching Policy: Network-First
- API Architecture Style: REST
- HTTP Client: Dio
- Authentication: Remote
- Biometric: FALSE
- Data Encryption: TRUE
- UI Rendering: Cross-Platform Widgets
- Navigation: Stack
- Theme Support: Light Theme
- UI Testing: FALSE
- Camera: FALSE
- Location Services: FALSE
- Push Notifications: FALSE
- Domain: Retail
- Product Catalog: TRUE
- Shopping Cart: TRUE

- Product Detail: FALSE

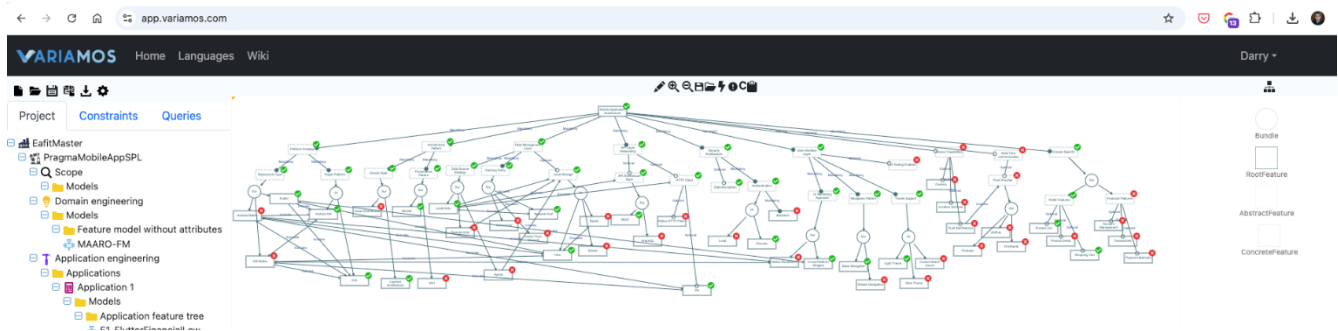


Figura 5. Configuración E3 en VariaMos

- E4: Aplicación Flutter para dominio *retail* de alta complejidad:
  - Deployment Type: Flutter
  - Target Platform: Android = TRUE, iOS = TRUE
  - Overall Style: Clean Architecture
  - Presentation Pattern: MVI
  - Data Source Strategy: Hybrid
  - Local Storage: Hive # ajuste correcto
  - Caching Policy: Cache-Then-Network
  - API Architecture Style: GraphQL
  - HTTP Client: Dio
  - Authentication: Remote
  - Biometric: TRUE
  - Data Encryption: TRUE
  - UI Rendering: Cross-Platform Widgets
  - Navigation: Drawer
  - Theme Support: Light Theme, Dark Theme, Custom Brand Colors
  - UI Testing: TRUE
  - Camera: TRUE
  - Location Services: TRUE
  - Push Notifications: TRUE
  - Push Provider: Firebase Cloud Messaging
  - Domain: Retail
  - Product Catalog: TRUE

- Shopping Cart: TRUE
- Product Detail: TRUE

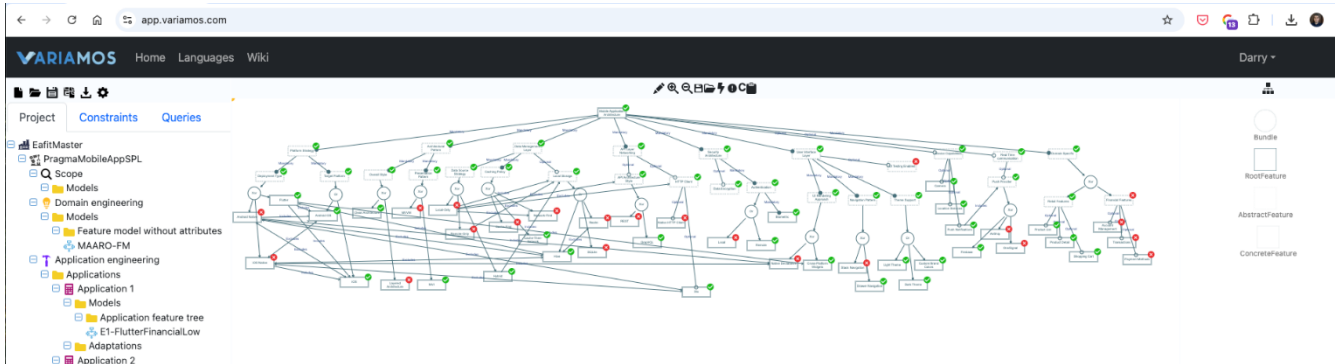


Figura 6. Configuración E4 en VariaMos

- E5: Aplicación Android nativa para dominio financiero de media complejidad:
  - Deployment Type: Android Native
  - Target Platform: Android
  - Overall Style: Clean Architecture
  - Presentation Pattern: MVVM
  - Data Source Strategy: Hybrid
  - Local Storage: SQLite
  - Caching Policy: Network-First
  - API Architecture Style: REST
  - HTTP Client: Native HTTP Client
  - Authentication: Remote
  - Biometric: FALSE
  - Data Encryption: TRUE
  - UI Rendering: Native Declarative
  - Navigation: Stack
  - Theme Support: Light Theme
  - UI Testing: FALSE
  - Camera: FALSE
  - Location Services: FALSE
  - Push Notifications: True
  - Push Provider: Firebase Cloud Messaging
  - Domain: Financial

- Account Management: TRUE
- Transactions: TRUE
- Payment Methods: FALSE

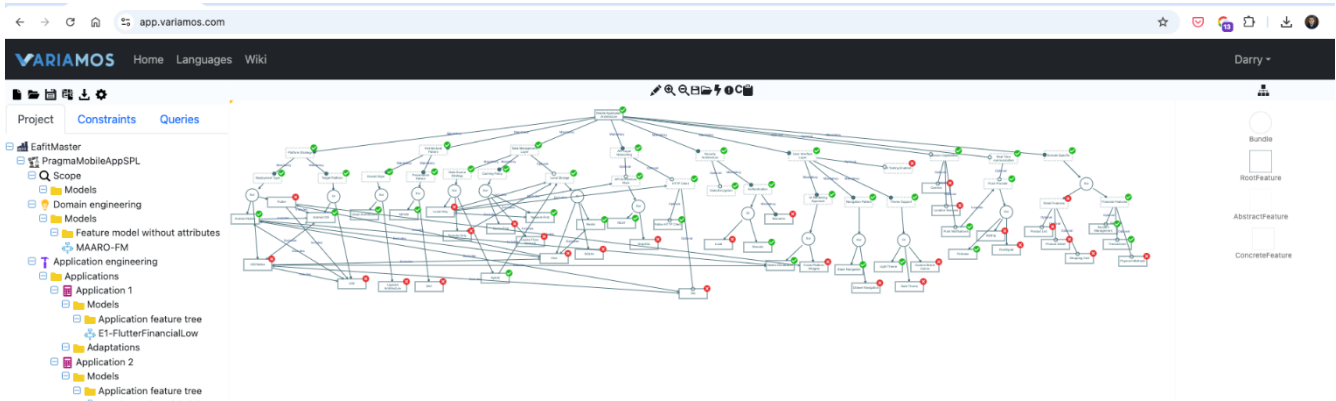


Figura 7. Configuración E5 en VariaMos

- E6: Aplicación iOS nativa para dominio financiero de media complejidad:
  - Deployment Type: iOS Native
  - Target Platform: iOS
  - Overall Style: Clean Architecture
  - Presentation Pattern: MVVM
  - Data Source Strategy: Hybrid
  - Local Storage: SQLite
  - Caching Policy: Network-First
  - API Architecture Style: REST
  - HTTP Client: Native HTTP Client
  - Authentication: Remote
  - Biometric: FALSE
  - Data Encryption: TRUE
  - UI Rendering: Native Declarative
  - Navigation: Stack
  - Theme Support: Light Theme
  - UI Testing: FALSE
  - Camera: FALSE
  - Location Services: FALSE
  - Push Notifications: True

- Push Provider: Firebase Cloud Messaging
- Domain: Financial
- Account Management: TRUE
- Transactions: TRUE
- Payment Methods: FALSE

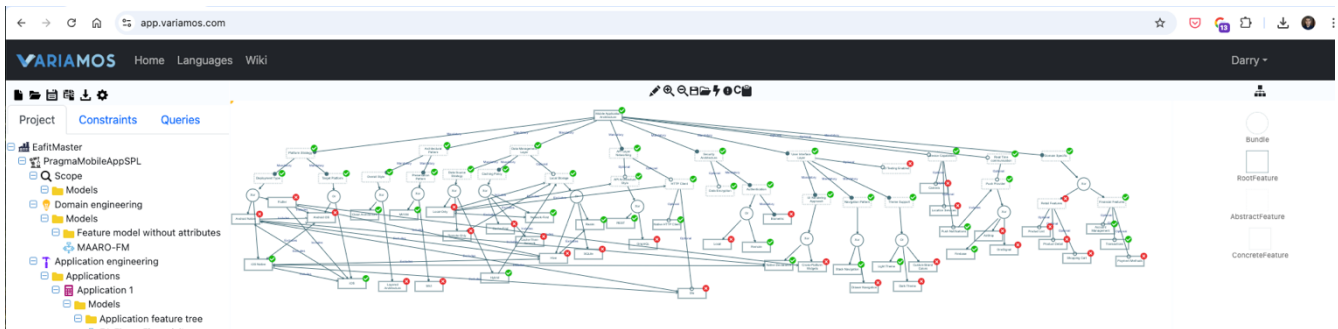


Figura 8. Configuración E6 en VariaMos

- Se desarrollaron los procesos automatizados para la generación de proyectos bases a partir de la configuración de cada escenario:
  - Para proyectos basados en Flutter, se utilizó la dependencia Mason para generar componentes reutilizables o *bricks*, que puedan ser importados bajo demanda desde los procesos de *backend* de VariaMos o ejecutados vía scripts a partir de la definición de la configuración de VariaMos (`config.json`), una vez cada configuración estuviera validada.
    - Se generaron diversos *bricks* para cubrir la variabilidad del modelo de características reducido: `core_project_brick`, `architecture_brick`, `data_brick`, `feature_brick`.
    - `Core_project_brick`: incluyó la definición inicial del Proyecto base, incluyendo la definición de dependencias base, manejo de inyección de dependencias general, definición del *theme* general, constantes generales, ambientes de compilación.
    - `Architecture_brick`: incluyó la estructura de carpetas inicial, el patrón de arquitectura (limpia o por capas), la definición base para inyección de dependencias, la configuración de las dependencias y el manejador de estados particular.
    - `Data_brick`: incluyó la configuración de persistencia y red, fuentes de datos locales, remotas o híbridas, estilo de api, manejador local de datos y políticas de

- chaché, manejo de abstracciones, cliente de api según el estilo definido y las implementaciones de los contratos de comunicación.
- Feature\_brick: incluyó módulos específicos por dominios *retail* y financiero, estructura de carpeta del modelo de características, entidades, repositorios, casos de uso, modelos, páginas y navegación básica.
  - Estos bricks lograron cubrir hasta un aproximado de 80% de las posibles variaciones del modelo operacional reducido presentado.
- Para proyecto basados en tecnologías nativas para móviles (Kotlin, Swift)
    - Se creó scripts de Gradle para Android, con la generación de *templates* kotlin que cubre la variabilidad indicada por el modelo reducido hasta en un 70%: estructura de carpetas, manejo de dependencias, gestión de permisos de usuarios. El script consideró los mismos criterios iniciales tomados en cuenta desde el generador Flutter basado en Mason: *Core Project task*, *Architecture task*, *Data task* y *Feature Domain task*.
    - Se crearon scripts de XCodegen para Android, con la generación de *templates* Swift que cubre la variabilidad indicada por el modelo reducido hasta en un 70%: estructura de carpetas, manejo de dependencias, gestión de permisos de usuarios. El script consideró los mismos criterios iniciales tomados en cuenta desde el generador Flutter basado en Mason: *Core Project Layer*, *Architecture Layer*, *Data Layer* y *Feature Domain Layer*.
    - Dichas automatizaciones significaron, en alcance, menos que Flutter que cuenta con herramientas más especializadas para estos casos de generación automática como Mason.

#### 5.3.4. FASE 4: EXPERIMENTACIÓN CONTROLADA

Objetivo específico asociado: Comparar y evaluar el proceso actual de derivación versus el proceso basado en líneas de productos de software, midiendo el impacto en la calidad del proyecto base y la eficiencia de derivación.

Actividades (4 semanas):

Para la ejecución del cuasi-experimento planteado se realizaron las siguientes etapas:

1. Preparación del experimento

- a. Se generó un documento con las especificaciones de los requisitos correspondiente a cada escenario 1-6, destacando requisitos de arquitectura solicitados, características propias del dominio indicado y restricciones a nivel de tecnología.
  - b. Se estableció el uso de las versiones de tecnologías y librerías más recientes al momento del inicio del experimento.
  - c. Se realizó una sesión de *onboarding* con los ingenieros participantes para contextualizar el *checklist* CRM para la verificación posterior y resolución de dudas o ambigüedades.
  - d. Se configuraron los proyectos privados de Github de la organización para asegurar el manejo de Git *hooks*, activación de pipeline con Github *actions*, integración con SonarCloud.
  - e. Se asignó un ingeniero de desarrollo *mobile* con 4 años de experiencia (Flutter) para las derivaciones de los escenarios 1-4.
  - f. Se asignó un ingeniero de desarrollo *mobile* con 4 años de experiencia (nativo) para las derivaciones de los escenarios 5-6.
  - g. Se solicitó a cada ingeniero derivar tanto manualmente como de forma semi-automatizada con LPS.
  - h. Se determinó un orden de ejecución alternada entre escenarios y métodos y un tiempo máximo de 2 días entre derivación y derivación, quedando de esta forma:
    - i. E1: primero, manual y luego semi-automatizado.
    - ii. E2: primero bajo LPS y luego manual
    - iii. E3: primero de forma manual y posteriormente semi-automatizado con LPS.
    - iv. E4: primero mediante LPS para continuar con derivación manual.
    - v. E5: en forma manual y luego bajo LPS.
    - vi. E6: se ejecutó vía LPS para cerrar con el proceso manual.
2. Ejecución de las derivaciones manuales de los escenarios 1-6.
- a. Cada iteración sobre los escenarios planteados se realizaron los siguientes pasos:
    - i. Se entregó al ingeniero a cargo la especificación del escenario junto con el enlace al repositorio de código que debía usar.
    - ii. Se generó el primer *commit* que representó la marca *t\_inicial*.
    - iii. Se ejecutó el proceso de derivación manual desde cero, con pausas naturales permitidas.
    - iv. Al momento de solicitud de completitud por parte del ingeniero, se registró el tiempo referencial.

- v. Se realizó una evaluación utilizando el *checklist* CRM para determinar completitud, en caso de que no se cumpliera 100% el ingeniero a cargo continuó con su labor hasta alcanzarlo.
  - vi. Cuando se alcanzó el 100% cubierto del CRM, se capturó el  $t_{\text{final}}$  desde el *commit* final al repositorio y se determina el tiempo de derivación manual:  $t_{\text{final}} - t_{\text{inicial}}$ .
3. Ejecución de las derivaciones mediante línea de productos de software de los escenarios 1-6.
- a. Se realizó una contextualización de 30min con cada ingeniero para visualizar las capacidades de la herramienta VariaMos y el modo de uso esperado.
  - b. Se realizó una contextualización de las herramientas de generación automática de código a partir de las tecnologías puntuales: Mason (Flutter), Gradle (Android), XCodegen (iOS)
  - c. Cada iteración sobre los escenarios planteados se realizaron los siguientes pasos:
    - i. Se registró el  $t_{\text{inicial}}$  al instante de inicio de sesión en la herramienta VariaMos.
    - ii. Se ejecutó el proceso de derivación semi-automática desde cero, a partir del feature modelo público MAAFM-RO, tomando en cuenta fueron permitidas pausas naturales.
    - iii. El ingeniero exportó el archivo config.json de la configuración indicada en el escenario y luego utilizó la herramienta, particular indicado para su escenario (Mason, Gradle, XCodegen), que permitió generar el proyecto base esperado.
    - iv. El ingeniero complementó las mejoras o ajustes del código marcadas como TODO, en el proyecto base generado.
    - v. Al momento de solicitud de completitud por parte del ingeniero, se registró el tiempo referencial.
    - vi. Se realizó una evaluación utilizando el checklist CRM para determinar completitud, en caso de que no se cumpliera 100% el ingeniero a cargo continuó con su labor hasta alcanzarlo.
    - vii. Cuando se alcanzó el 100% cubierto del CRM, se capturó el  $t_{\text{final}}$  desde el *commit* final al repositorio y se determina el tiempo de derivación vía LPS:  $t_{\text{final}} - t_{\text{inicial}}$ .
4. Evaluación heurística de la calidad de los proyectos base derivados
- a. Los proyectos base generados se etiquetaron como P01 hasta P12, independientemente del método de derivación usado.

- b. Los evaluadores no conocían de contexto cual proyecto había sido generado por cual método de derivación.
- c. Los evaluadores trabajaron de forma independiente, se asignaron 2 evaluadores por cada proyecto o escenario que probar.
- d. Cada evaluador completo el formulario de valoración heurística bajo escala Likert.
- e. Se consolidaron los resultados por proyecto y se calculó el valor IQ\_Heurística.
- f. Los proyectos que fueron valorados con una diferencia mayor a 10 puntos fueron enviados a una sesión de consenso.
- g. Se calculó el IQ-Arq del proyecto, a partir del IQ\_SonaCloud y del IQ\_Heurística y se consolidaron los datos en una tabla.

## 5. Fase de análisis de los resultados

### a. Preparación de los datos

- i. Se alistó un documento .csv en el cual se consolidó el *set* de datos por cada ejecución de derivación, resaltando la información referida a: ProyectoID, EscenarioID, Tecnología, Dominio, NivelComplejidad, MétodoDerivación, SeniorityDev, FechaI, FechaF, TiempoDerivaciónMin, IQ\_ARQ, IQ\_SonarCloud, IQ\_Heurística, MétricasSonarQube: (Bugs/KLOC, Vulnerabilidades/KLOC, CodeSmells/KLOC, ComplejidadCiclomática, PorcentajeCobertura, LíneasDuplicadas), CRMFinal, CRMNumCiclos, Notas.
- ii. Se realizó una validación de datos que contempló:
  - 1. Ningún dato de los planteados podía faltar.
  - 2. Los tiempos siempre expresados en minutos.
  - 3. Los tiempos debían englobar todas las fases hasta el cumplimiento 100% de la valoración CRM.
- iii. Se ejecutó el cálculo de las diferencias pareadas por cada escenario, en cuanto a tiempo (TIEMPO\_DERIVACION) y eficiencia (IQ\_ARQ):
  - 1.  $\Delta\_TIEMPO\_DERIVACION\_i = TIEMPO\_DERIVACION\_SPL\_i - TIEMPO\_DERIVACION\_MANUAL\_i$
  - 2.  $\Delta\_IQ\_ARQ\_i = IQ\_ARQ\_SPL\_i - IQ\_ARQ\_MANUAL\_i$

### b. Análisis estadístico

- i. Análisis de variable IQ\_ARQ:

1. Se ejecutó la prueba de verificación de normalidad Shapiro-Wilk sobre  $\Delta\_IQ\_ARQ$  para la obtención de las variables W y p-valor.
  2. Se realizó la prueba **t de student** para la muestra pareada unilateral, considerando los grados de libertad  $n= 6$  escenarios  $-1 = 5$  y nivel de significancia  $\alpha=0,05$ .
  3. Se calculó el tamaño del efecto por medio de la d de Cohen para la muestra pareada.
  4. Se calculó el intervalo de confianza del promedio de las diferencias basado en un re-muestreo de 10000.
- ii. Análisis de variable TIEMPO\_DERIVACION:
1. Se ejecutó la prueba de verificación de normalidad **Shapiro-Wilk** sobre  $\Delta\_TIEMPO\_DERIVACION$  para la obtención de las variables W y p-valor.
  2. Se realizó la prueba **t de student** para la muestra pareada unilateral, considerando los grados de libertad  $n= 6$  escenarios  $-1 = 5$  y nivel de significancia  $\alpha=0,05$ .
  3. Se calculó el tamaño del efecto por medio de la **d de Cohen** para la muestra pareada.
  4. Se calculó el intervalo de confianza del promedio de las diferencias basado en un re-muestreo de 10000.
- c. Análisis cualitativo de casos
- i. Se seleccionaron 2 escenarios representativos para un análisis cualitativo llevado a cabo con un arquitecto de aplicaciones móviles.
  - ii. El criterio de selección fue aquellos escenarios que presentaron un mayor y menor ahorro en tiempo de derivación.
  - iii. Se inspeccionó el resultado de los proyectos base generados en ambos escenarios para terminar hallazgos cualitativos frente a: métricas generadas, estándares de calidad y buenas prácticas de desarrollo.
  - iv. Se documentan los hallazgos encontrados durante la inspección de un lapso de 4 horas.
- d. Visualización de datos

Los datos fueron agrupados de manera que generaron los siguientes gráficos que facilitaron la discusión de los resultados:

- i. Gráfico de barras agrupadas con la variable de TIEMPO\_DERIVACION, tanto manual como con SPL, para cada uno de los escenarios ejecutados.
- ii. Gráfico de barras agrupadas con la variable IQ\_ARQ, tanto manual como con prácticas SPL, para cada uno de los escenarios ejecutados.

#### 5.4.CONSIDERACIONES ÉTICAS

Esta investigación planteó algunos principios éticos que se mencionan a continuación:

- Consentimiento informado: todas las personas participantes fueron informados debidamente del objetivo de la investigación y su participación fue voluntaria y sin recolección de datos sensibles.
- Confidencialidad: los nombres de proyectos fueron anonimizados, el código fuente y métricas de estos fueron almacenados en repositorios privados.
- Conflicto de interés: el investigador del estudio forma parte de la empresa Pragma S.A, lo cual podría derivar en un sesgo hacia resultados positivos. Esto se mitiga bajo un estudio estadístico de rigor y valoraciones heurísticas por terceras personas.

## 6. PRESUPUESTO

El presupuesto de esta investigación se conformó por las siguientes categorías, donde se desglosan los elementos que la componen y su costo asociado dependiendo del contexto de este:

### 1. Salarios

- a. Ingeniero de aplicaciones móviles
  - i. Honorarios mes USD \$1500 jornada completa
  - ii. Disponibilidad media jornada durante 6 meses = **USD \$4500**
- b. Arquitecto de aplicaciones móviles
  - i. Honorarios hora USD \$25
  - ii. Disponibilidad 10 horas mes durante 4 meses = **USD \$1000**
- c. Consultor de Ingeniería de líneas de productos.
  - i. Honorarios hora USD \$20
  - ii. Disponibilidad 20 horas mes durante 5 meses = **USD \$2000**
- d. Consultor de Calidad de Software
  - i. Honorarios hora USD \$7
  - ii. Disponibilidad 20 horas mes durante 3 meses = **USD \$420**

### 2. Herramientas de software y licencias

- a. SonarCloud
  - i. Plan Team USD \$30 por usuario mes
  - ii. Disponibilidad de 2 usuarios por 6 meses = **USD \$360**
- b. Software de modelado de líneas de productos VARIAMOS
  - i. Plan gratuito USD = **0\$**
- c. Github
  - i. Plan Team USD \$4 por usuario mes
  - ii. Disponibilidad de 5 usuarios por 6 meses = **USD \$120**

### 3. Servicios Cloud (Azure)

- a. Azure Container Apps
  - i. Plan pago por uso USD \$72 por actividad 7x24 mes con 1 millón de peticiones
  - ii. Disponibilidad 7x24 durante 4 meses = aproximadamente **USD \$288**

Total, presupuesto por la duración de 6 meses = **USD \$7788**

## 7. RESULTADOS

### 7.1. EFICIENCIA DE DERIVACIÓN (TIEMPO\_DERIVACIÓN)

La Tabla I a continuación presenta los tiempos de derivación medidos, durante la fase de experimentación, para cada uno de los escenarios que se contemplaron y ejecutaron.

Tabla I. TIEMPO DE DERIVACIÓN OBSERVADOS (TIEMPO EFECTIVO)

Escenario	Tecnología	Dominio	Complejidad	TIEMPO_DERIVACIÓN		Reducción	
				Manual	SPL	min	%
E1	Flutter	Retail	Baja	2,434	1,288	1,146	47.10%
E2	Flutter	Retail	Alta	5,297	2,367	2,930	55.30%
E3	Flutter	Financial	Baja	2,387	1,270	1,117	46.80%
E4	Flutter	Financial	Alta	5,401	2,447	2,954	54.70%
E5	Android	Retail	Media	3,969	2,013	1,956	49.30%
E6	iOS	Retail	Media	4,198	2,275	1,923	45.80%

El promedio del tiempo de derivación manual fue 3,948 minutos

El promedio del tiempo de derivación bajo SPL fue 1,9483 minutos

Con el método de SPL se registró una reducción promedio del tiempo de derivación de 49.8%

#### 7.1.1. DISTRIBUCIÓN DE TIEMPOS POR COMPLEJIDAD

Según se detalla en la Tabla II, los proyectos que tienen una complejidad más alta presentan una reducción porcentual del tiempo de derivación, a partir del uso de SPL, también más alta. Esto puede deberse a que a mayor cantidad de características también aumenta el código *boilerplate*, que termina siendo reemplazado por componentes reusables al momento de la generación semi-automática.

Tabla II. DISTRIBUCIÓN DE TIEMPOS DE DERIVACIÓN POR COMPLEJIDAD DE ESCENARIO

Complejidad	n (pares)	TIEMPO_DERIVACIÓN Promedio		Reducción Promedio	
		Manual	SPL	min	%
Baja	2	2,411	1,279	1,132	46.94%
Media	2	4,084	2,144	1,940	47.50%
Alta	2	5,349	2,407	2,942	55.00%

### 7.1.2. DISTRIBUCIÓN DE TIEMPOS POR TECNOLOGÍA

En la Tabla III se puede entender que la mayor reducción promedio porcentual se dio bajo la tecnología de Flutter con un 52%, lo cual es consistente con el mayor alcance que tenía el generador automatizado con Mason. En cambio, iOS mostró la menor de las reducciones promedio de tiempo de derivación dado lo limitado que puede ser el generador utilizando XCodegen y las fricciones técnicas propias de las restricciones del ecosistema de Apple. Pese a esto, la diferencia entre todas las tecnologías no supera el los 7%, lo que sugiere que el beneficio de las prácticas SPL es igualmente robusto en todas las plataformas estudiadas para esta variable

Tabla III. DISTRIBUCIÓN DE TIEMPOS DE DERIVACIÓN POR TECNOLOGÍA

Tecnología	n (pares)	TIEMPO_DERIVACIÓN Promedio		Reducción Promedio	
		Manual	SPL	min	%
Flutter	4	3,880	1,843	2,037	52.50%
Android	1	3,969	2,013	1,956	49.28%
iOS	1	4,198	2,275	1,923	45.81%

### 7.1.3. GRÁFICO DE TIEMPOS DE DERIVACIÓN

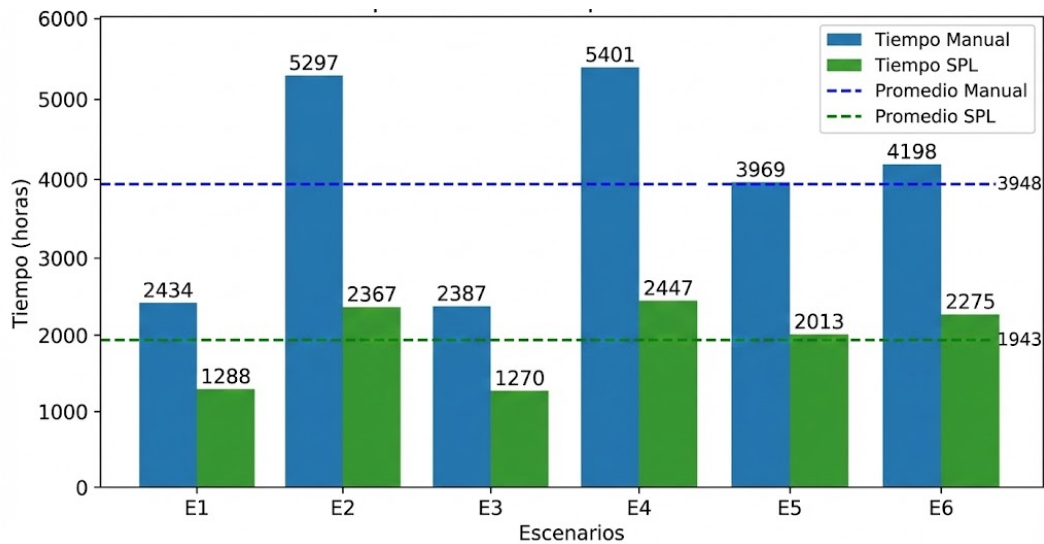


Figura 9. Tiempo de derivación por escenario y modalidad.

La gráfica anterior muestra la disminución de los tiempos de derivación usando las prácticas de SPL de forma consistente en todos los escenarios, representando una disminución promedio entorno al 49%.

## 7.2. CALIDAD DE PROYECTO BASE (IQ\_ARQ)

La tabla IV presenta los resultados totalizados de calidad del proyecto base según la operacionalización de planteada de la variable IQ\_ARQ, mostrando las diferencias registradas por método.

Tabla IV. CALIDAD DE PROYECTO BASE (IQ\_ARQ)

Escenario	Método	IQ_SonarCloud	IQ_Heurística	IQ_Arq	Aumento de IQ_ARQ	
					puntos	%
E1	Manual	61.3	58.7	60.2	-	-
E1	SPL	85.7	76.8	81.8	+21.6	35.91%
E2	Manual	51.2	59.5	54.8	-	-
E2	SPL	75.8	66.9	72	+17.2	31.44%
E3	Manual	60.2	59	59.7	-	-
E3	SPL	86.9	75	81.6	+21.9	36.72%

E4	Manual	48.5	57.2	52.1	-	-
E4	SPL	72.1	64.8	68.9	16.8	32.25%
E5	Manual	54.8	58.3	56.3	-	-
E5	SPL	77.3	70.2	74.2	+17.9	31.80%
E6	Manual	53.2	59.2	55.8	-	-
E6	SPL	75.8	69.5	73.1	+17.3	31.04%

El cálculo de IQ\_ARQ se generó aplicando:

$$IQ\_ARQ = (0.60 * IQ\_SONARCLOUD) + (0.40 * IQ\_HEURÍSTICA)$$

El cálculo de la variación se generó por medio de:

$$\Delta IQ\_ARQ\_Ei = IQ\_ARQ\_SPL\_Ei - IQ\_ARQ\_MANUAL\_Ei$$

El porcentaje de mejora se calculó usando

$$\% IQ\_ARQ\_Ei = (\Delta IQ\_ARQ\_Ei / IQ\_ARQ\_MANUAL\_Ei) * 100$$

El resultado promedio de IQ\_SONARCLOUD tuvo una mejora cercana al 44% mientras que el IQ\_HEURÍSTICA presentó un aumento del 20.3%, lo que, en conjunto, significó que el valor de IQ\_ARQ bajo LPS aumentara alrededor de 19 puntos, representando una mejor respecto al método manual de aproximadamente 33%

### 7.2.1. DESAGREGACIÓN DE IQ\_SONARCLOUD

En la Tabla V se muestran los valores inversos de las métricas operacionalizadas de esta forma, junto con normalización de los datos como cobertura y duplicados de código, necesarios para obtener el promedio ponderado indicado previamente para esta variable.

Tabla V. DESAGREGACIÓN DE IQ\_SONARCLOUD

Métrica	Rango de Referencia	Promedio		Mejora (%)	Observación
		Manual	SPL		
Bugs/KLOC	< 1.0 (excelente)	2.33	1.05	-54.90%	El método con SPL redujo bugs a la mitad
Vulnerabilidades/KLOC	< 0.5 (excelente)	0.95	0.25	-73.70%	El método con SPL disminuyó vulnerabilidades críticas
Code Smells/KLOC	< 5.0 (excelente)	12.93	5.83	-54.90%	El método con SPL redujo hallazgos de mantenibilidad
Complejidad Ciclomática	< 4.0 (excelente)	5.02	2.83	-43.60%	El método con SPL produjo funciones menos complejas
Cobertura de Código (%)	> 80 (excelente)	20.70%	50.20%	+142.50%	El método con SPL incluye los <i>tests</i> base automáticamente
Líneas de código duplicadas (%)	< 3.0 (excelente)	8.72%	2.63%	-69.80%	El método con SPL impulsa la reutilización

Entre los hallazgos claves se pudo identificar una mejora pronunciada (descenso) sobre las vulnerabilidades con 73% y líneas de código duplicadas con 69% menos respecto al método manual. La explicación sobre el aumento de 142% sobre el valor detectado de cobertura de código puede deberse a que los scripts de los generadores semiautomáticos ya incluyeron algunos sets básicos de pruebas unitarias sobre los componentes sugeridos de código.

### 7.2.2. DESAGREGACIÓN DE IQ\_HEURISTICA

En la Tabla VI se muestra el resultado obtenido, de forma segregado, para la IQ\_HEURISTICA, tomando en cuenta la calificación que otorgó cada valorador a los 16 *items* que la componen (4 atributos de calidad y características asociadas), esto bajo escala Likert 1-5.

Tabla VI. DESAGREGACIÓN DE IQ\_HEURÍSTICA

Atributo ISO 25010	Promedio		Mejora	
	Manual	SPL	puntos	%
Mantenibilidad	62.5	78.1	+15.6	25.00%
Fiabilidad	56.3	68.8	+12.5	22.20%
Seguridad	54.7	66.9	+12.2	22.30%
Portabilidad	61.2	68.4	+7.2	11.80%
<b>Promedio General</b>	58.6	70.5	+11.9	20.30%

La puntuación por evaluador se obtuvo siguiendo:

$$IQ\_HEURISTICA\_EVALUADOR\_i = \Sigma (Puntuación\_item\_i) \text{ para } i = 1..16$$

Se ejecutó tanto para los proyectos derivados manualmente como con SPL. Para determinar la heurística final se usó:

$$IQ\_HEURISTICA = Promedio (IQ\_HEURISTICA\_EVALUADOR\_1 + IQ\_HEURISTICA\_EVALUADOR\_2)$$

El coeficiente de correlación intraclase que verificó el acuerdo entre evaluadores se situó en 0.84, cuya interpretación es de acuerdo “excelente”

### 7.2.3. DISTRIBUCIÓN DE CALIDAD POR COMPLEJIDAD

Los proyectos de baja complejidad mostraron una mejora relativamente mejor que el resto, tal y como se aprecia en la Tabla VII con la distribución de calidad por complejidad de cada escenario.

Tabla VII. DISTRIBUCIÓN DE CALIDAD IQ\_ARQ POR COMPLEJIDAD DE ESCENARIO

Complejidad	n (pares)	IQ_ARQ Promedio		Mejora	
		Manual	SPL	puntos	%
Baja	2	60	81.7	+21.7	36.20%
Media	2	56.1	73.7	+17.6	31.40%
Alta	2	53.5	70.5	+17	31.80%

En contraste con esto se tiene que los proyectos de alta complejidad requieren de decisiones más complejas por lo que al momento de generar los proyectos base no se logró obtener la misma proporción de avance. Sin embargo, en todos los escenarios de complejidad se observó resultados positivos de aumento de IQ\_ARQ promedio.

#### 7.2.4. GRÁFICO DE CALIDAD DE PROYECTO BASE

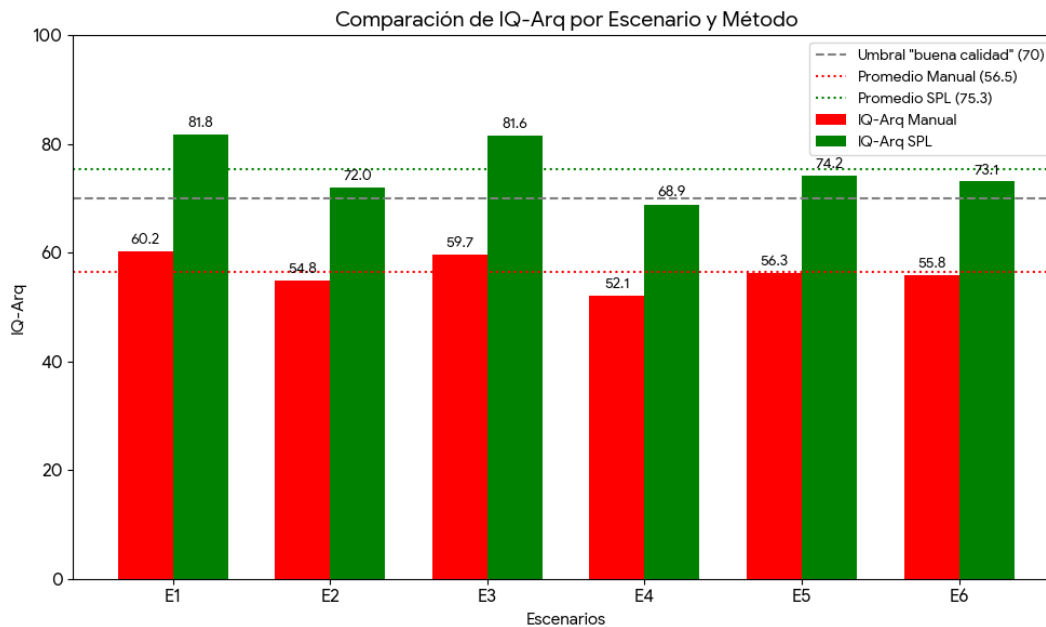


Figura 10. Calidad del proyecto base derivado por escenario.

El criterio de calidad de proyecto base derivado (IQ\_ARQ), como se aprecia en la imagen, superó consistentemente, bajo el método de SPL, la línea base definida como umbral de buena calidad situada en 70 de 100. Hubo mejoras de esta variable en todos los escenarios, aunque de manera menos significativa en proyectos relativos a desarrollo nativo como Android e iOS, comparando directamente con Flutter (E1-E4)

### 7.3. ANÁLISIS ESTADÍSTICO INFERENCIAL

A continuación, se estableció los cálculos llamados a determinar estadísticamente las diferencias que se observaron para la derivación manual y bajo SPL y su significancia.

#### 7.3.1. ANÁLISIS DE DIFERENCIAS PAREADAS

Inicialmente se calculó las diferencias de cada par (derivación manual y derivación con SPL) para cada variable dependiente planteada en el experimento, para cada uno de los 6 escenarios planteados, tal como se aprecia en la Tabla VIII.

Tabla VIII. DIFERENCIAS PAREADAS OBSERVADAS EN EL EXPERIMENTO

Escenario	$\Delta$ _Tiempo_derivacion (min)	$\Delta$ _IQ-Arq (puntos)
E1	-1,146	21.6
E2	-2,930	17.2
E3	-1,117	21.9
E4	-2,954	16.8
E5	-1,956	17.9
E6	-1,923	17.3

#### Análisis del $\Delta$ \_TIEMPO\_DERIVACIÓN:

Media:  $\bar{D}$ \_TIEMPO\_DERIVACIÓN =  $\Sigma (\Delta_i) / n = -2004.3$  minutos

Desviación estándar:  $S_d = [ \Sigma ( \Delta_i - \bar{D} )^2 / ( n - 1 ) ]$

$$S_d = \sqrt{(3,291,773 / 5)} = \sqrt{658,355} = \mathbf{811.4 \text{ minutos}}$$

### Análisis del $\Delta_{IQ\_ARQ}$ :

Media:  $\bar{D}_{IQ\_ARQ} = \Sigma (\Delta_i) / n = 18.78$  puntos

Desviación estándar:  $S_d = [ \Sigma ( \Delta_i - \bar{D} )^2 / ( n - 1 ) ] = 2.19$  puntos

#### 7.3.2. PRUEBA DE NORMALIDAD SHAPIRO-WILK

El experimento requirió la determinación del supuesto de normalidad de las diferencias pareadas para poder realizar la prueba t de Student.

Las hipótesis para este caso fueron:

- $H_0$ : Las diferencias pareadas siguen una distribución normal
- $H_1$ : Las diferencias pareadas no siguen una distribución normal.

El nivel de significancia establecido fue  $\alpha = 0.05$

Prueba de normalidad para  $\Delta_{TIEMPO\_DERIVACIÓN}$

Datos = [ -1146, -2930, -1117, -2954, -1956, -1923 ]

Estadístico  $W = 0.794$

p-valor = 0.058

$p = 0.058 > \alpha = 0.05 \rightarrow$  No se rechazó la  $H_0$ .

$\Delta_{TIEMPO\_DERIVACIÓN}$  tuvo una distribución normal.

Prueba de normalidad para  $\Delta_{IQ\_ARQ}$

Datos = [ 21.6, 17.2, 21.9, 16.8, 17.9, 17.3 ]

Estadístico  $W = 0.885$

p-valor = 0.299

$p = 0.299 > \alpha = 0.05 \rightarrow$  No se rechazó la  $H_0$ .

$\Delta_{TIEMPO\_DERIVACIÓN}$  tuvo una distribución normal.

### 7.3.3. PRUEBA T DE STUDENT PAREADA (TIEMPO\_DERIVACIÓN)

Las hipótesis para este caso fueron:

- $H_0$ : La derivación usando SPL no reduce tiempo de derivación.
  - $H_0: \mu_{\text{Tiempo\_SPL}} \geq \mu_{\text{Tiempo\_Manual}}$
- $H_1$ : La derivación usando SPL reduce tiempo de derivación.
  - $H_1: \mu_{\text{Tiempo\_SPL}} < \mu_{\text{Tiempo\_Manual}}$

El nivel de significancia establecido fue  $\alpha = 0.05$

$$t = \bar{D} / (S_d / \sqrt{n})$$

Datos previamente calculados:

$$\bar{D} = -2,004.3 \text{ min}$$

$$S_d = 811.4 \text{ min}$$

$$n = 6$$

$$t = \bar{D} / (S_d / \sqrt{n}) = -6.05$$

con grados de libertad  $df = n-1 = 5$

El valor  $t_{\text{crítico}}$  fue ( $\alpha = 0.05$ ,  $df = 5$ , unilateral izquierda) = -2.015

$$t_{\text{calculado}} = t = -6.05 < t_{\text{crítico}} = -2.015$$

$|t_{\text{calculado}}| > |t_{\text{crítico}}| \Rightarrow$  Se rechazó la  $H_0$ .

Se calculó el p-valor unilateral =  $P(T \leq -6.05 \mid df = 5)$

$$p\text{-valor} = 0.0008 < \alpha = 0.05$$

Se concluyó por evidencia estadística ( $t = -6.05$  y  $p\text{-valor} < 0.001$ ) rechazar la hipótesis nula, por lo cual se entendió que el tiempo de derivación a partir de SPL es significativamente menor que el tiempo de derivación manual en Pragma.

### 7.3.4. PRUEBA T DE STUDENT PAREADA (IQ\_ARQ)

Las hipótesis para este caso fueron:

- $H_0$ : La derivación usando SPL no mejora la calidad del proyecto base derivado.
  - $H_0: \mu_{IQ-Arq\_SPL} \leq \mu_{IQ-Arq\_Manual}$
- $H_1$ : La derivación usando SPL mejora la calidad del proyecto base derivado.
  - $H_1: \mu_{IQ-Arq\_SPL} > \mu_{IQ-Arq\_Manual}$

El nivel de significancia establecido fue  $\alpha = 0.05$

$$t = \bar{D} / (S_d / \sqrt{n})$$

Datos previamente calculados:

$$\bar{D} = +18.78 \text{ min}$$

$$S_d = 2.19 \text{ min}$$

$$n = 6$$

$$t = \bar{D} / (S_d / \sqrt{n}) = \mathbf{21.00}$$

con grados de libertad  $df = n-1 = 5$

El valor  $t_{\text{crítico}}$  fue ( $\alpha = 0.05$ ,  $df = 5$ , unilateral izquierda) = **+2.015**

$t_{\text{calculado}} = t = 21.00 > t_{\text{crítico}} = 2.015 \Rightarrow$  Se rechazó la  $H_0$

Se calculó el p-valor unilateral =  $P(T \leq 21.00 \mid df = 5)$

p-valor =  $0.0001 < \alpha = 0.05$

Se concluyó por evidencia estadística ( $t = 21.00$ ,  $p < 0.001$ ) rechazar la hipótesis nula, por lo cual se entendió que la calidad del proyecto base a partir de SPL es significativamente superior que la calidad del proyecto base manual en Pragma.

### 7.3.5. TAMAÑO DEL EFECTO COHEN'S D PAREADA

Calcular el tamaño del efecto por medio de la  $d$  de Cohen permitió evaluar la magnitud práctica de las diferencias observadas en la experimentación.

$$d = |\bar{D}| / S_d$$

Análisis de tiempo de derivación:

$$d_{\text{Tiempo\_Derivación}} = | -2,004.3 | / 811.4 = \mathbf{2.47}$$

$| d_{\text{Tiempo\_Derivación}} | \geq 1.3 \rightarrow$  se confirmó efecto muy grande, se rechazó  $H_0$ .

Análisis de calidad de proyecto base:

$$d_{\text{IQArq}} = 18.78 / 2.19 = \mathbf{8.58}$$

$d_{\text{IQArq}} \geq 1.3 \rightarrow$  se confirmó efecto muy grande, se rechazó  $H_0$ .

En ambos casos previos se demostró un efecto muy grande ( $d > 2.0$ ) con una significancia estadística muy fuerte ( $p > 0.001$ ) por lo cual se aseguró:

- Las diferencias detectadas entre ambos métodos de derivación son reales y no responden a temas del azar.
- La magnitud de las diferencias es sustancial, desde una perspectiva estadística práctica empresarial.
- Aunque la muestra  $n=6$  se consideró como pequeña, los efectos fueron tan significativos que se logró detectar con un nivel alto de confiabilidad.
- El poder estadístico de  $n=6$  es limitado para detectar efectos pequeños ( $d < 0.5$ ). Sin embargo, los efectos calculados son tan significativos ( $d > 2.0$ ) que el diseño del experimento tiene poder  $>95\%$  para detectarlos [39].

### 7.3.6. INTEVALOS DE CONFIANZA

El análisis estadístico se complementó con una prueba de intervalos de confianza Bootstrap (no paramétricos) para robustecer los hallazgos.

Se completó el siguiente algoritmo para cada variable ( $\Delta_{\text{TIEMPO\_DERIVACIÓN}}$ ,  $\Delta_{\text{IQ\_ARQ}}$ ):

- Se inicializó el arreglo de muestras
- Para cada  $i$  con  $i=1..10,000$ :
  - Remuestrear  $n = 6$  diferencias pareadas con reemplazo
  - Calcular la media de la muestra:  $\bar{D}_i$
  - Agregar  $\bar{D}_i$  al arreglo.

- Ordenar arreglo
- IC 95% = [ Percentil 2.5 , Percentil 97.5 ]

### Bootstrap para $\Delta$ \_TIEMPO\_DERIVACIÓN

IC 95 % = [ -2,847, -1,203 ] minutos

La reducción promedio de tiempo de derivación con SPL disminuyo entre 1,203 y 2,847 minutos con un 95% de confianza estadística.

### Bootstrap para $\Delta$ \_IQ\_ARQ

IC 95 % = [ +16.9, +20.8 ] puntos

La reducción promedio calidad del proyecto base con SPL aumentó entre 16.9 y 20.8 puntos con un 95% de confianza estadística.

Tabla IX. INTERVALOS DE CONFIANZA 95% (PARAMÉTRICO Y BOOTSTRAP)

Variable	IC 95% Paramétrico (t)	IC 95% Bootstrap	Convergencia
$\Delta$ _Tiempo_Derivación (min)	[ -2,821, -1,188 ]	[ -2,847, -1,203 ]	Alta
$\Delta$ _IQ-Arq (puntos)	[ +17.0, +20.6 ]	[ +16.9, +20.8 ]	Alta

De acuerdo con estos resultados de la Tabla IX se concluyó que los intervalos paramétricos y bootstrap son muy similares entre sí, validando así la tendencia a distribución normal de los datos y por ende la confianza en los resultados obtenidos.

#### 7.3.7. PODER ESTADÍSTICO POST-HOC

Se calculó el poder estadístico alcanzado en el experimento  $(1-\beta)$  dado el tamaño de efecto calculado y el numero de la muestra  $n=6$ .

- TIEMPO\_DERIVACIÓN ( $d = 2.47$  ;  $\alpha = 0.05$ ) Poder  $\approx 97.3\%$  : La probabilidad de no detectar efecto real en el experimento fue sólo de 2.7% mientras que hubo 97.3% de probabilidad de detectar un efecto de magnitud grande  $d = 2.47$ .

- IQ\_ARQ ( $d = 8.58$  ;  $\alpha = 0.05$ ) Poder  $\approx 99\%$  : La probabilidad de no detectar efecto real en el experimento fue sólo de 1% mientras que hubo 99% de probabilidad de detectar un efecto de magnitud grande  $d = 8.58$ .

### 7.3.8. ANÁLISIS CUALITATIVO DE CASOS

#### 7.3.8.1. CASO A: ESCENARIO 2

El caso analizado fue el proyecto base derivado para aplicación de *retail*, de alta complejidad, basado en Flutter. Con 12 características activas dentro del modelo operacional reducido. Representó el caso con mayor reducción de tiempo de derivación con una disminución del 55.3%.

#### **Resultados cuantitativos:**

- Tiempo Manual: 5,297 min
- Tiempo con SPL: 2,367 min
- Reducción: 2,930 min (55.3%)
- IQ-Arq Manual: 54.8
- IQ-Arq SPL: 72.0 (+12 puntos => 31.4%)

#### **Hallazgos significativos:**

- La estructura física del proyecto base manual presentó un mayor nivel de profundidad (9) con relación a los 7 niveles del proyecto base con SPL. Un mayor nivel de profundidad se puede entender como mayor complejidad a la mantenibilidad.
- La consistencia con las nomenclaturas o convenciones de nombramiento se adecuó totalmente a Dart en el proyecto base con SPL, mientras que en la derivación manual se presentan inconsistencias de hasta 3 estilos diferentes detectados.
- La separación de las capas de arquitectura fue más estricta en el proyecto con derivación SPL, manteniendo interfaces (contratos) de comunicación entre capas, mientras que el proyecto derivado manualmente se detectó inconsistencias como acoplamiento con clases concretas en lugar de abstracciones que limita la testeabilidad.
- La gestión de errores fue mejor particularizada en la derivación con SPL con definición de clases especializadas para modelado de errores.

```

28 // lib/repositories/product_repository.dart
29 class ProductRepository {
30   final http.Client client; // Acomplimiento
31   Future<List<Product>> getProducts() async {
32     // Hardcoding
33     final response = await client.get(
34       Uri.parse('https://api.example.com/products'),
35     );
36     if (response.statusCode == 200) {
37       // Ausencia de modelos
38       return (json.decode(response.body) as List)
39         .map((item) => Product.fromJson(item))
40         .toList();
41     }
42     throw Exception('Error'); // Gestión genérica de errores
43   }
44 }
45
46
47
48
49
50 // lib/features/product_catalog/domain/repositories/product_repository.dart
51 abstract class ProductRepository {
52   Future<Either<Failure, List<Product>>> getProducts();
53 }
54
55 // lib/features/product_catalog/data/repositories/product_repository_impl.dart
56 class ProductRepositoryImpl implements ProductRepository {
57
58   // Dependencia de abstracciones
59   final ProductRemoteDataSource remoteDataSource;
60   final ProductLocalDataSource localDataSource;
61   final NetworkInfo networkInfo;
62
63   // Inyección de dependencias
64   ProductRepositoryImpl({
65     required this.remoteDataSource,
66     required this.localDataSource,
67     required this.networkInfo,
68   });
69
70   @override
71   Future<Either<Failure, List<Product>>> getProducts() async {
72     // Verificación de la conectividad de red
73     if (await networkInfo.isConnected) {
74       try {
75         final remoteProducts = await remoteDataSource.getProducts();
76         localDataSource.cacheProducts(remoteProducts);
77         return Right(remoteProducts);
78       } on ServerException {
79         return Left(ServerFailure()); //Gestión de errores particular
80       }
81     }
82   }
83 }

```

Figura 11. Comparativa de código entre derivación manual y derivación con SPL

## Conclusión del caso:

El proyecto derivado por SPL tuvo influencia directa sobre las capas de dominio y data principalmente, apoyado de definiciones claras pre-validadas. Sin embargo, las características complejas del dominio requirieron del esfuerzo manual del ingeniero a cargo para satisfacer decisiones de negocio que los scripts de generación no pueden anticipar. El 55.3% de mejora en tiempo de derivación reflejó la reducción de las tareas más repetitivas del andamiaje del proyecto, pero no necesariamente del contenido de negocio del dominio.

### 7.3.8.2. CASO B: ESCENARIO 6

El caso analizado fue el proyecto base derivado para aplicación de Retail, de complejidad media, basado en iOS nativo. Con 8 *features* activas dentro del modelo operacional reducido. Representó el caso con menor reducción de tiempo de derivación con una disminución del 45.8%.

## Resultados cuantitativos:

- Tiempo Manual: 4,198 min
- Tiempo SPL: 2,275 min
- Reducción: 1,923 min (45.8%)
- IQ-Arq Manual: 55.8
- IQ-Arq SPL: 73.1 (+17.3 puntos => 31.0%)

## Hallazgos significativos:

- Existió una fricción técnica compartida para ambos modelos de derivación referente a la gestión de los certificados y perfiles de aprovisionamiento, gestión de dependencias CocoaPods y ralentización de proceso de *building*, todos propios del ecosistema de desarrollo de Apple.
- Los scripts de generación para la plataforma de iOS desde la derivación SPL no tuvo el mismo alcance ni profundidad que el obtenido con Flutter, razón por lo cual la mejor no significó un porcentaje similar, aun considerando una complejidad funcional más simple.
- La arquitectura resultante del proyecto derivado con SPL mostró una separación de responsabilidades rigurosa mientras que en el proyecto con derivación manual presentó inconsistencias especialmente para el manejo de patrones de inyección de dependencias.
- El proyecto con derivación manual presentó hallazgos sobre manejo de prácticas de programación orientada a protocolos que no fueron detectados en el proyecto de derivación con SPL.
- El ecosistema de desarrollo más restringido como Apple limita las posibilidades de generación automatizada y de abstracción.

## Conclusión del caso:

Efectivamente se registró una mejora en el tiempo de derivación de 45%, menor al del caso anterior, pero manteniendo una valoración de calidad 31% cercana al obtenido con Flutter con 34%, lo que significó una mejora de 17 puntos sobre esta variable en comparación con la derivación manual. Esto valida el enfoque de portabilidad de la línea de productos incluyendo alcances bajo tecnologías nativas.

## 7.4. RESÚMEN DE LOS RESULTADOS EXPERIMENTALES

La Tabla X muestra el resumen de los hallazgos determinados con este estudio, de los cuales tomó relevancia la reducción de tiempos de derivación bajo el proceso semi-automático que usó mecanismos de SPL para la derivación de proyectos base para el desarrollo de apps móviles, reduciendo un tiempo promedio de 49.8%. Por su parte, la variable de calidad de código bajo el proceso de derivación con SPL presentó mejora promedio equivalente del 33.2%.

Tabla X. SÍNTESIS DE HALLAZGOS PRINCIPALES

Aspecto	Hallazgo	Evidencia
Reducción de tiempo de derivación	La derivación con SPL reduce tiempo de derivación promedio en 49.8% (2,004 min)	t = -6.05, p < 0.001 d = 2.47
Rango de reducción de tiempo de derivación	Entre 45.8% (iOS - Media) y 55.3% (Flutter-Alta)	IC 95%: [47.5%, 52.5%] bootstrap
Mejora de calidad de proyecto base	La derivación con SPL aumenta IQ-ARQ promedio en 33.2% (18.8 puntos)	t = 21.00 p < 0.0001 d = 8.58
Rango de mejora de calidad de proyecto base	Entre 31.0% (iOS - Media) y 36.7% (Flutter-Baja)	IC 95%: [+16.9, +20.8] bootstrap

## 7.5. VALIDACIÓN DE HIPOTESIS

### H1: Calidad de proyecto del base

H<sub>0</sub>:  $\mu_{IQ\_ARQ\_SPL} \leq \mu_{IQ\_ARQ\_Manual}$

H<sub>1</sub>:  $\mu_{IQ\_ARQ\_SPL} > \mu_{IQ\_ARQ\_Manual}$ . (con  $\alpha=0.05$ )

Se rechazó la hipótesis H<sub>0</sub> y se confirmó la hipótesis H<sub>1</sub>. El proceso de derivación apoyado por SPL mejora significativamente la calidad de los proyectos base derivados, con un nivel de confianza superior a 99%, con datos estadísticos que lo soportan ( t = 21.00, p < 0.0001, d = 8.58 ).

### H2: Eficiencia de derivación

H<sub>0</sub>:  $\mu_{TIEMPO\_DERIVACION\_SPL} \geq \mu_{TIEMPO\_DERIVACION\_Manual}$

H<sub>1</sub>:  $\mu_{TIEMPO\_DERIVACION\_SPL} < \mu_{TIEMPO\_DERIVACION\_Manual}$ . (con  $\alpha=0.05$ )

Se rechazó la hipótesis H<sub>0</sub> y se confirmó la hipótesis H<sub>1</sub>. El proceso de derivación apoyado por LPS disminuye significativamente el tiempo de derivación, con un nivel de confianza superior a 99% y datos estadísticos que lo soportan ( t = -6.05, p < 0.001, d = 2.47)

Basado en esta evidencia empírica, se afirma que la implementación de una línea de productos de software para el proceso de derivación de proyectos genera un impacto positivo muy significativo para las ambas dimensiones evaluadas:

- Eficiencia del proceso de derivación: con reducciones promedios de 49.8% con evidencia estadística muy fuerte ( $p < 0.001$ ) y efecto extremadamente alto ( $d = 2.47$ ).
- Calidad arquitectónica: con mejoras promedios de 33.2% según el criterio IQ\_ARQ empleado en la experimentación, pasando de nivel de calidad aceptable a buena. Esto con base estadística extremadamente fuerte ( $p < 0.0001$ ) y efecto extremadamente grande ( $d = 8.58$ ).
- Los beneficios fueron consistentes independientemente de las tecnologías evaluadas (Flutter, Android, iOS), dominios (*Reatil, Financial*) y complejidad de los escenarios (baja, media, alta).

La adopción de LPS en Pragma. S.A es viable desde el punto de vista técnico y positivamente válida desde el punto de vista estadístico. Su aporte para reducir tiempos de ingeniería y mejorar la calidad de software en la derivación de proyectos base móviles es muy significativa a partir de las evidencias de este estudio.

## 7.6. DISCUSIÓN DE RESULTADOS

### 7.6.1. INTERPRETACION DE LA REDUCCIÓN DE TIEMPOS Y MEJORA EN CALIDAD

La evidencia derivada de los resultados de estudio sugiere mejoras en los tiempos de derivación cerca al 50% y aumento promedio en la calidad de los proyectos base derivados equivalente a 33%. Sin embargo, es importante reconocer que esto no se traduce directamente en una idea de ineficiencia directa del proceso manual, sino debe entenderse que las prácticas de la ingeniería de LPS reduce o eliminan sistemáticamente tiempos muertos y carga cognitiva en los ingenieros que las aplican.

El proceso de derivación manual está sujeto a la alternancia entre toma de decisiones arquitectónicas y las implementaciones de las mismas, quedando expuesto a síntomas de fatiga cognitiva que pueden derivar en pausas o merma de productividad. La derivación mediante LPS, en cambio, encapsula estas decisiones en activos pre-validados y por tanto la ganancia detectada en la experimentación no proviene de unos procesos de codificación más acelerado sino de la eliminación de la re-inversión de la estructura base del proyecto inicial, reducción del error humano y de ciertas actividades inherentes al proceso creativo de desarrollo artesanal.

## 7.6.2. ANÁLISIS DEL RETORNO DE INVERSIÓN ROI Y PUNTO DE EQUILIBRIO

Los hallazgos demostraron la efectividad de las prácticas de SPL para el proceso de derivación de proyectos base, considerando tiempo de derivación y calidad del proyecto generado, sin embargo, se debe contemplar también que pese a los resultados positivos, para que este proceso termine siendo escalable y sostenible en el tiempo se debe velar por rentabilidad, lo cual guarda relación directa con el costo en la adopción de una línea de productos de software, considerando el importe inicial por la ejecución de la fase de ingeniería de dominio para la determinación de la variabilidad correcta para el negocio, y la el desarrollo del ecosistema de artefactos o generadores pre-validados que se requiere para dar vida útil a la LPS.

Al considerar el ahorro promedio en horas calculado en un proceso de derivación y tomando en cuenta el costo (en horas) que representaría el montaje de toda una LPS (lenguajes abstractos, lenguaje concreto, modelado de dominio, generación de aplicación, modelamiento de restricciones) sería necesario la evaluación del punto de equilibrio para determinar en qué lapso podría conseguirse un retorno de la inversión inicial. También se hace necesario resaltar que para este estudio se logró capitalizar herramientas como VariaMos y se refinó uno modelo operacional reducido que acotara la cantidad de artefactos y tiempo de construcción de estos.

## 8. CONCLUSIONES

Este estudio tuvo como intención principal determinar cómo las prácticas de ingeniería de líneas de producto de software impactan la calidad del proyecto base y la eficiencia de su derivación para el desarrollo de aplicaciones móviles en Pragma S.A. Con base en los resultados obtenidos, fue posible sintetizar esta serie de conclusiones que responden directamente a los objetivos planteados:

- Se determinó que la ingeniería de líneas de productos de software genera un impacto positivo y estadísticamente significativo, reduciendo el tiempo de derivación en 49.8% mientras aumenta la calidad arquitectónica un 33.2% frente al método de derivación tradicional usando actualmente en Pragma S.A.
- Se determinó que el proceso de derivación manual actual requiere un promedio de 3948 minutos para generar un proyecto base funcional, cuya calidad de código es siempre variable y dependiente del criterio individual del desarrollador y de las actividades inherentes de su rol.
- Se diseñó el modelo de características *mobile application architecture feature model - reduced operational*, con 72 puntos de variabilidad, que permitió validar la factibilidad de estandarizar la arquitectura de aplicaciones móviles heterogéneas (Flutter, Android, iOS) bajo un mismo modelo incluyendo características base de dominios financiero y *retail*.
- La derivación semi-automática mediante la integración con VariaMos y generadores de código específicos (Mason, Gradle, XCodegen) generó proyectos base exitosos abarcando hasta un 80% de la variabilidad modelada para Flutter y 70% para tecnologías nativas.
- La eficiencia del proceso de derivación con SPL es sensible al contexto tecnológico y a la calidad de los artefactos (scripts de generación, *templates*, clases) pre-validados, y pudo percibirse en la reducción de tiempos de 45.8% y 52.5% en Flutter e iOS respectivamente.

## 9. RECOMENDACIONES

Derivadas del análisis realizado y de las limitaciones identificadas, se presentan las recomendaciones que pueden guiar futuros trabajos e implementar mejoras prácticas:

Para la organización Pragma S.A:

- **Institucionalización:** la adopción práctica de la ingeniería de productos de software requiere de una intención clara y manifiesta dado que impactaría diversas áreas de la organización como desarrollo, devops y seguridad que deben habilitar y orquestar en conjunto la posibilidad de materializar inicio de nuevos proyectos de aplicaciones móviles bajo un nuevo estándar más significativo desde el punto de vista de eficiencia y calidad en el proceso una verdadera industrialización del software [49].
- **Mantenimiento de SPL y ROI:** se recomienda la creación de un comité especializado en el soporte evolutivo de la línea de productos de software, sus modelos de características inherentes y los diversos generadores necesarios para los casos contemplados en desarrollo *mobile*. Estos recursos deben estar siempre atendidos, mantenidos y soportados para la estabilidad de los resultados, por lo que debe considerarse un estudio de factibilidad económico que derive en la proyección de ROI a corto o mediano plazo.
- **Capacitación:** La transferencia de conocimiento acerca del diseño y uso de SPL en procesos de desarrollo de software es importante para reducir la curva de aprendizaje inicial detectada en la fase de experimentación.

Para futuros trabajos:

- **Ampliación de modelo operacional reducido:** se recomienda extender el modelo de características operacional reducido tomando como marco de referencia general el *mobile application architecture feature model* anexo en este estudio, que incluye también otros dominios de negocio.
- **SPL en conjunto con GenAI:** se recomienda explorar la integración de modelos de lenguajes LLMs con capacidades generativas de código para lograr cubrir ese 20-30% de variabilidad que está por fuera del alcance de los generadores estáticos sugeridos en este estudio.
- **Estudio largo plazo:** se recomienda medir no solamente etapas iniciales de proyectos sino identificar el impacto en mantenibilidad y deuda técnica en proyectos a largo plazo (+12 meses).

## 10. REFERENCIAS

- [1] A. Rocha, F. J. García Peñalvo, G. Paiva. Dias, and Ramiro. Gonçalves, “In-Depth Analysis of Mobile Apps Statistics: A Study and Development of a Mobile App,” in *in Proc. 18th Iberian Conf. Inf. Syst. Technol*, Jun. 2023.
- [2] G. S. . Tomar and J. Chand. Bansal, “Review on Mobile App Development: Tools and Techniques,” in *IEEE World Conf. Appl. Intell. Comput*, 2023.
- [3] A. M. Gutiérrez-Fernández, A. E. Chacón-Luna, D. Benavides, L. Fuentes, and R. Rabiser, “Variability management and software product line knowledge in software companies,” *J. Syst. Softw.*, vol. 216, Oct. 2024, doi: 10.1016/j.jss.2024.112114.
- [4] A. E. C. Luna, A. M. G. Fernandez, D. Benavides, and L. Fuentes, “Design for the analysis of variability management in the industry,” in *Proc. 26th ACM Int. Syst. Softw. Product Line Conf.*, Sep. 2022, pp. 117–122. doi: 10.1145/3503229.3547060.
- [5] M. Dahlem, R. Rahm, and M. Becker, “App variants and their impact on mobile architecture: An experience report,” in *Proc. ACM Int. Conf.*, Sep. 2019. doi: 10.1145/3336294.3336320.
- [6] K. C. Kang, S. Cohen, W. E. Novak, and J. A. Hess, “Feature-Oriented Domain Analysis (FODA) feasibility study,” Trevor Ashe, 1990.
- [7] K. Lee, K. C. Kang, and J. Lee, “Concepts and Guidelines of Feature Modeling for Product Line Software Engineering,” 2002. doi: 10.1007/3-540-46020-9\_5.
- [8] K. C. Kang, J. Lee, and P. Donohoe, “Feature-Oriented Product Line Engineering,” *IEEE Softw*, 2002.
- [9] A. Jaaksi, “Developing Mobile Browsers in a Product Line,” *IEEE Softw*, pp. 73–80, Jul. 2002, doi: 10.1109/MS.2002.1020290.
- [10] D. Muthig, I. John, and K. Schmid, “GoPhone-A Software Product Line in the Mobile Phone Domain,” *Fraunhofer IESE*, Mar. 2004, [Online]. Available: <https://www.researchgate.net/publication/240023161>
- [11] F. Bachmann and P. C. Clements, “Variability in Software Product Lines,” 2005.
- [12] J. Lee, D. Muthig, and M. Naab, “An approach for developing service oriented product lines,” in *Proc. 12th Int. Softw. Product Line Conf.*, 2008, pp. 275–284. doi: 10.1109/SPLC.2008.34.
- [13] J. Kim, S. Park, and V. Sugumaran, “DRAMA: A framework for domain requirements analysis and modeling architectures in software product lines,” *J. Syst. Softw.*, vol. 81, no. 1, pp. 37–55, Jan. 2008, doi: 10.1016/j.jss.2007.04.011.
- [14] J. Lee, D. Muthig, and M. Naab, “A feature-oriented approach for developing reusable product line assets of service-based systems,” *J. Syst. Softw.*, vol. 83, no. 7, pp. 1123–1136, Jul. 2010, doi: 10.1016/J.JSS.2010.01.048.
- [15] M. E. Joorabchi, A. Mesbah, and P. Kruchten, “Real challenges in mobile app development,” in *Proc. Int. Symp. Empir. Softw. Eng. Meas.*, 2013, pp. 15–24. doi: 10.1109/ESEM.2013.9.
- [16] Ricardo Erikson V. S. Rosa and Vicente F. Lucena Jr, “Smart composition of reusable software components inmobile application product lines,” in *Proc. 2nd Int. Workshop Product Line Approaches Softw. Eng.*, 2013. doi: 10.1145/1985484.1985496.
- [17] L. Pavlič, “The Mobile Product Line,” *SQAMIA*, 2013, [Online]. Available: <http://ceur-ws.org>
- [18] J. M. Vanhatupa, “Process model for rapid implementation of features using flexible architecture,” in *Proc. ACM Int. Conf.*, Association for Computing Machinery, Aug. 2015, pp. 27–31. doi: 10.1145/2785592.2785614.
- [19] N. Gámez, L. Fuentes, and J. M. Troya, “Creating Self-Adapting Mobile Systems with Dynamic Software Product Lines,” *IEEE Softw*, Mar. 2015.
- [20] H. Bagheri, J. Garcia, A. Sadeghi, S. Malek, and N. Medvidovic, “Software architectural principles in contemporary mobile software: from conception to practice,” *J. Syst. Softw.*, vol. 119, pp. 31–44, Sep. 2016, doi: 10.1016/j.jss.2016.05.039.
- [21] M. Usman, M. Z. Iqbal, and M. U. Khan, “A product-line model-driven engineering approach for generating feature-based mobile applications,” *J. Syst. Softw.*, vol. 123, pp. 1–32, Jan. 2017, doi: 10.1016/j.jss.2016.09.049.
- [22] T. Durschmid, M. Trapp, and J. Dollner, “Towards Architectural Styles for Android App Software Product Lines,” in *Proc. 2017 IEEE/ACM 4th Int. Conf. Mobile Softw. Eng. Syst.*, Jul. 2017, pp. 58–62. doi: 10.1109/MOBILESoft.2017.12.
- [23] J. F. Bastos, P. A. da Mota Silveira Neto, P. O’Leary, E. S. de Almeida, and S. R. de Lemos Meira, “Software product lines adoption in small organizations,” *J. Syst. Softw.*, vol. 131, pp. 112–128, Sep. 2017, doi: 10.1016/j.jss.2017.05.052.
- [24] E. M. Grua, I. Malavolta, and P. Lago, “Self-adaptation in mobile apps: A systematic literature study,” in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng. Adapt. Self-Managing Syst.*, May. 2019, pp. 51–62. doi: 10.1109/SEAMS.2019.00016.
- [25] W. K. G. Assunção, S. R. Vergilio, and R. E. Lopez-Herrejon, “Automatic extraction of product line architecture and feature models from UML class diagram variants,” *Inf Softw Technol*, vol. 117, Jan. 2020, doi: 10.1016/j.infsof.2019.106198.

- [26] M. Nieke, G. Sampaio, T. Thüm, C. Seidl, L. Teixeira, and I. Schaefer, "Guiding the evolution of product-line configurations," *Softw Syst Model*, vol. 21, no. 1, pp. 225–247, Feb. 2022, doi: 10.1007/s10270-021-00906-w.
- [27] S. Khoshnevis and O. Ardestani, "Search-based approaches to optimizing software product line architectures: A systematic literature review," *Inf. Softw. Technol.*, Jun. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.infsof.2024.107446.
- [28] L. Pavlič, T. Beranič, and M. Heričko, "A product quality impacts of a mobile software product line: an empirical study," *PeerJ Comput Sci*, vol. 7, pp. 1–26, Apr. 2021, doi: 10.7717/peerj-cs.434.
- [29] D. Benavides, S. Segura, and A. Ruiz-Cortés, "Automated analysis of feature models 20 years later: A literature review," *Inf Syst*, vol. 35, no. 6, pp. 615–636, Sep. 2010, doi: 10.1016/j.is.2010.01.001.
- [30] S. Apel, D. Batory, C. Kästner, and G. Saake, "Feature-Oriented Software Product Lines," Springer, 2013. doi: 10.1007/978-3-642-37521-7.
- [31] A. Hevner, S. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75-105, 2004.
- [32] J. W. Creswell and V. L. Plano Clark, *Designing and Conducting Mixed Methods Research*, 3rd ed. Thousand Oaks, CA: SAGE Publications, 2018.
- [33] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77, 2007.
- [34] R. K. Yin, *Case Study Research and Applications: Design and Methods*, 6th ed. Thousand Oaks, CA: SAGE Publications, 2018.
- [35] W. R. Shadish, T. D. Cook, and D. T. Campbell, *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*, Boston, MA: Houghton Mifflin, 2002.
- [36] M. Q. Patton, *Qualitative Research & Evaluation Methods*, 4th ed. Thousand Oaks, CA: SAGE Publications, 2015.
- [37] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131-164, 2009.
- [38] ISO/IEC, "ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuRE)," International Organization for Standardization, 2011.
- [39] V. B. Kampenes, T. Dybå, J. E. Hannay, and D. I. K. Sjøberg, "A systematic review of effect size in software engineering experiments," *Information and Software Technology*, vol. 49, no. 11-12, pp. 1073-1086, 2007.
- [40] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.
- [41] M. B. Miles, A. M. Huberman, and J. Saldaña, *Qualitative Data Analysis: A Methods Sourcebook*, 3rd ed. Thousand Oaks, CA: SAGE Publications, 2014.
- [42] M. Jørgensen, "A review of studies on expert estimation of software development effort," *Journal of Systems and Software*, vol. 70, no. 1-2, pp. 37-60, 2004.
- [43] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Software Engineering Institute, Carnegie Mellon University, Technical Report CMU/SEI-90-TR-21, 1990.
- [44] J. Bosch, "Designing Software Product Lines for Industrial Applications," in "Software Product Lines: Proceedings of the Second International Conference, SPLC 2", San Diego, CA, USA: Springer, 2000, pp. 33-50.
- [45] P. Clements and L. Northrop, "Software Product Lines: Practices and Patterns". Boston, MA, USA: Addison-Wesley, 2002.
- [46] K. Pohl, G. Böckle, and F. van der Linden, "Software Product Line Engineering: Foundations, Principles, and Techniques". Berlin, Germany: Springer, 2005.
- [47] F. van der Linden, K. Schmid, and E. Rommes, "Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering". Berlin, Germany: Springer, 2007.
- [48] B. A. Kitchenham et al., "Preliminary guidelines for empirical research in software engineering," *IEEE Trans. Software Eng.*, vol. 28, no. 8, pp. 721-734, 2002.
- [49] R. Mazo, *Guía para la adopción industrial de líneas de productos de software*. Medellín, Colombia: Editorial EAFIT, 2018. ISBN: 978-958-720-506-0.

## 11. ANEXOS

### 11.1. EVALUACIÓN HEURÍSTICA

#### ANEXO A: PROTOCOLO DE EVALUACIÓN HEURÍSTICA ISO 25010

##### Instrucciones para el evaluador:

1. Clone el repositorio del proyecto a evaluar
2. Revise README y documentación arquitectónica (10 min)
3. Inspeccione código: navegue estructura, revise 3 clases por capa (25 min)
4. Complete el formulario siguiente (15 min)

**IMPORTANTE:** NO se le informará si el proyecto es Manual o SPL hasta completar todas las evaluaciones (cegamiento).

##### FORMULARIO DE EVALUACIÓN HEURÍSTICA

**Proyecto ID:** \_\_\_\_\_ (código anónimo asignado)

**Evaluador:** \_\_\_\_\_

**Fecha:** \_\_\_\_\_

**ESCALA LIKERT:** 1 = Muy Pobre | 2 = Pobre | 3 = Aceptable | 4 = Bueno | 5 = Excelente

#### 1. MANTENIBILIDAD

Subcaracterística	Criterio de Evaluación	Puntuación (1-5)	Nota
1.1 Modularidad	¿Las responsabilidades están bien separadas en módulos cohesivos con bajo acoplamiento?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
1.2 Reusabilidad	¿Existen componentes genéricos reutilizables (no acoplados a lógica específica)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
1.3 Analizabilidad	¿La estructura del código es clara y navegable? ¿Nombres de clases/funciones son descriptivos?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
1.4 Modificabilidad	¿Qué tan fácil sería agregar una nueva feature sin modificar código existente?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	

**Puntuación Mantenibilidad:** \_\_\_\_/20

## 2. FIABILIDAD

Subcaracterística	Criterio de Evaluación	Puntuación (1-5)	Nota
<b>2.1 Madurez</b>	¿El código parece completo y bien implementado (sin TODOs críticos, funciones vacías)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<b>2.2 Tolerancia a fallos</b>	¿Existen mecanismos de manejo de errores (try-catch, validaciones)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<b>2.3 Recuperabilidad</b>	¿El código tiene estrategias de recuperación ante fallos (retry, fallback, offline-first)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<b>2.4 Disponibilidad</b>	¿El diseño facilita alta disponibilidad (ej. caché, persistencia local)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	

**Puntuación Fiabilidad:** \_\_\_\_/20

## 3. SEGURIDAD

Subcaracterística	Criterio de Evaluación	Puntuación (1-5)	Nota
<b>3.1 Confidencialidad</b>	¿Los datos sensibles están encriptados (en reposo y en tránsito)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<b>3.2 Integridad</b>	¿Hay validación de inputs? ¿Se previenen inyecciones (SQL, XSS)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<b>3.3 Autenticidad</b>	¿Existe gestión adecuada de identidades/autenticación?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<b>3.4 Responsabilidad</b>	¿Hay mecanismos de auditoría (logging de acciones críticas)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	

**Puntuación Seguridad:** \_\_\_\_/20

#### 4. PORTABILIDAD

Subcaracterística	Criterio de Evaluación	Puntuación (1-5)	Nota
4.1 Adaptabilidad	¿El código es independiente de detalles de plataforma (uso de abstracciones)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
4.2 Instalabilidad	¿Qué tan fácil es instalar y ejecutar el proyecto (README claro, dependencias bien gestionadas)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
4.3 Reemplazabilidad	¿Las dependencias externas están abstraídas (fácil cambiar librerías sin refactorización masiva)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
4.4 Coexistencia	¿El código sigue convenciones estándar de la tecnología (facilita colaboración)?	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	

**Puntuación Portabilidad:** \_\_\_\_/20

#### CONSOLIDACIÓN

**Puntuación Total:** \_\_\_\_/80 (suma de 4 atributos)

Normalización a escala 0-100:

Code

$$IQ\_Heurística\_Evaluador = [(Puntuación\ Total - 16) / 64] \times 100 = \underline{\hspace{2cm}}$$

#### COMENTARIOS CUALITATIVOS GENERALES

**Fortalezas arquitectónicas principales (máx 200 palabras):**

**Debilidades arquitectónicas principales (máx 200 palabras):**

**Recomendaciones de mejora (máx 150 palabras):**

**Firma del Evaluador:** \_\_\_\_\_

**Fecha de Evaluación:** \_\_\_\_\_

## 11.2. CHECKLIST DE REQUISITOS MINIMOS (CRM)

### ANEXO B: CHECKLIST DE REQUISITOS MÍNIMOS (CRM)

Instrucciones: Este checklist debe ser aplicado por un evaluador independiente (no el ingeniero que realizó la derivación) para determinar si el proyecto base alcanzó el estado de "completitud mínima" (CRM = 100%).

Proyecto ID: \_\_\_\_\_

Escenario: \_\_\_\_\_

Método de Derivación:  Manual  SPL

Ingeniero Responsable: \_\_\_\_\_

Evaluador CRM: \_\_\_\_\_

Fecha de Evaluación: \_\_\_\_\_

#### ÍTEMS DEL CHECKLIST (12 ítems binarios)

#	Ítem	Cumple	No Cumple	Método de Verificación
1	Compilación exitosa en CI/CD	<input type="checkbox"/>	<input type="checkbox"/>	Revisar reporte de GitHub Actions/GitLab CI: ¿Pipeline verde?
2	Ejecución local sin crashes	<input type="checkbox"/>	<input type="checkbox"/>	Ejecutar app en emulador: ¿Abre? ¿Navega entre $\geq 2$ pantallas? ¿No crashea?
3	Estructura modular arquitectónica	<input type="checkbox"/>	<input type="checkbox"/>	Inspeccionar árbol de directorios: ¿Existen carpetas según patrón declarado (ej. /domain, /data, /presentation)?
4	Configuración de entornos (Dev/Prod)	<input type="checkbox"/>	<input type="checkbox"/>	Buscar archivos: ¿Existen config.dev.* y config.prod.* o build flavors?
5	Gestión segura de secretos	<input type="checkbox"/>	<input type="checkbox"/>	Ejecutar: <code>grep -r "api_key password secret" lib/src/</code> → ¿Retorna 0 resultados literales?
6	Linting configurado (0 errores críticos)	<input type="checkbox"/>	<input type="checkbox"/>	Ejecutar flutter analyze, <code>./gradlew lint</code> , o <code>swiftlint</code> → ¿Pasa sin errores nivel ERROR?
7	Al menos 1 prueba unitaria ejecutable	<input type="checkbox"/>	<input type="checkbox"/>	Ejecutar flutter test, <code>./gradlew test</code> , o <code>xcodebuild test</code> → ¿Al menos 1 test pasa?
8	Logging básico implementado	<input type="checkbox"/>	<input type="checkbox"/>	Buscar uso de logger en $\geq 3$ archivos: <code>grep -r</code>

				"logger Log.d print" lib/ src/ → ¿Al menos 3 archivos?
9	Branding configurable	<input type="checkbox"/>	<input type="checkbox"/>	Revisar theme.dart, colors.xml, o Assets.xcassets: ¿Colores/nombre app configurables (no hardcoded en widgets)?
10	Documentación técnica README	<input type="checkbox"/>	<input type="checkbox"/>	Abrir README.md: ¿Contiene setup instructions, arquitectura, estructura carpetas?
11	Estrategia de ramas Git	<input type="checkbox"/>	<input type="checkbox"/>	Ejecutar git branch -a: ¿Existen ≥2 ramas (ej. main + develop)?
12	Auditoría de dependencias	<input type="checkbox"/>	<input type="checkbox"/>	Revisar SonarCloud o ejecutar flutter pub outdated: ¿Sin vulnerabilidades CRÍTICAS/HIGH?

## RESULTADO

Ítems Cumplidos: \_\_\_\_/12

CRM (%): \_\_\_\_%

Decisión:

CRM = 100% → Proyecto APROBADO. Registrar t\_final. Proceder a análisis de calidad.

CRM < 100% → Proyecto NO completado. Ingeniero debe corregir ítems faltantes y repetir evaluación.

ÍTEMS FALTANTES (si CRM < 100%)

Listar números de ítems que no cumplieron:

Notas para el ingeniero (qué debe corregir):

Firma del Evaluador: \_\_\_\_\_

Timestamp de Evaluación: \_\_\_\_\_

### 11.3. MOBILE APPLICATION ARCHITECTURE FEATURE MODEL – REDUCED OPERATIONAL (MAAFM-RO)

#### Modelo Operacional Reducido (MAAFM-OR) — v1.7 para la Experiencia Experimental

Propósito: Permitir la diferenciación y configuración automatizada de los 6 escenarios experimentales (Flutter, Android, iOS × Retail/Financial × Baja/Alta complejidad)

##### 1. Criterios de selección de features

- Discriminan entre escenarios experimentales (diferenciadoras)
- Son objetivamente verificables por checklist, análisis estático o inspección estructural
- Son implementables en los generadores Mason/scripts
- Cubren las principales categorías: Plataforma, Arquitectura, Persistencia, API, Seguridad, UI, Capabilities, Dominio
- Permiten experimentar los constraints y dependencias críticas del modelo conceptual

##### 2. Features seleccionados

Mobile Application Architecture [ROOT - MANDATORY]

###### --- 1. PLATFORM STRATEGY [MANDATORY]

- └─ Deployment Type [MANDATORY - XOR]
  - └─ Flutter [ALTERNATIVE]
  - └─ Android Native [ALTERNATIVE]
    - └─ iOS Native [ALTERNATIVE]
- └─ Target Platform [MANDATORY - OR]
  - └─ Android [OPTIONAL]
  - └─ iOS [OPTIONAL]

###### --- 2. ARCHITECTURAL PATTERN [MANDATORY]

- └─ Overall Style [MANDATORY - XOR]
  - └─ Clean Architecture [ALTERNATIVE]
    - └─ Layered Architecture [ALTERNATIVE]
- └─ Presentation Pattern [MANDATORY - XOR]
  - └─ MVVM [ALTERNATIVE]
  - └─ MVI [ALTERNATIVE]

###### --- 3. DATA MANAGEMENT LAYER [MANDATORY]

- └─ Data Source Strategy [MANDATORY - XOR]
  - └─ Local-Only [ALTERNATIVE]
  - └─ Remote-Only [ALTERNATIVE]
    - └─ Hybrid [ALTERNATIVE]
- └─ Local Storage [OPTIONAL - OR]
  - └─ SQLite [OPTIONAL] # Relational, multiplataforma
  - └─ Realm [OPTIONAL] # No Relational, multiplataforma
    - └─ Hive [OPTIONAL] # NoSQL, común en Flutter
- └─ Caching Policy [MANDATORY - XOR]
  - └─ Cache-First [ALTERNATIVE]

- └─ Network-First [ALTERNATIVE]
- └─ Cache-Then-Network [ALTERNATIVE]

--- 4. API LAYER NETWORKING [MANDATORY]

- └─ API Architecture Style [MANDATORY - XOR]
  - └─ REST [ALTERNATIVE]
  - └─ GraphQL [ALTERNATIVE]
- └─ HTTP Client [MANDATORY - XOR]
  - └─ Native HTTP Client [ALTERNATIVE]
  - └─ Dio [ALTERNATIVE]

--- 5. SECURITY ARCHITECTURE [MANDATORY]

- └─ Authentication [MANDATORY - OR]
  - └─ Local [OPTIONAL] # PIN/Password
  - └─ Remote [OPTIONAL] # JWT/OAuth
  - └─ Biometric [OPTIONAL] # Solo alta complejidad
- └─ Data Encryption [OPTIONAL]

--- 6. USER INTERFACE LAYER [MANDATORY]

- └─ UI Rendering Approach [MANDATORY - XOR]
  - └─ Native Declarative [ALTERNATIVE] # iOS SwiftUI, Android Jetpack Compose
  - └─ Cross-Platform Widgets [ALTERNATIVE] # Flutter widgets
- └─ Navigation Pattern [MANDATORY - XOR]
  - └─ Stack Navigation [ALTERNATIVE]
  - └─ Drawer Navigation [ALTERNATIVE]
- └─ Theme Support [MANDATORY - OR]
  - └─ Light Theme [OPTIONAL]
  - └─ Dark Theme [OPTIONAL]
  - └─ Custom Brand Colors [OPTIONAL]
- └─ UI Testing Enabled [OPTIONAL]

--- 7. DEVICE CAPABILITIES [OPTIONAL]

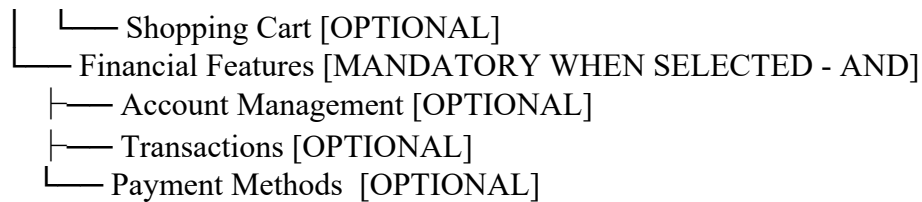
- └─ Camera [OPTIONAL]
- └─ Location Services [OPTIONAL]
- └─ Push Notifications [OPTIONAL]

--- 8. REAL-TIME COMMUNICATION [OPTIONAL]

- └─ Push Provider [XOR] # Mandatory if Push Notifications
  - └─ Firebase Cloud Messaging [ALTERNATIVE]
  - └─ AirShip [ALTERNATIVE]
  - └─ OneSignal [ALTERNATIVE]

--- 9. DOMAIN-SPECIFIC EXTENSION (MANDATORY - XOR)

- └─ Retail Features [MANDATORY WHEN SELECTED - AND]
  - └─ Product List [OPTIONAL]
  - └─ Product Detail [OPTIONAL]

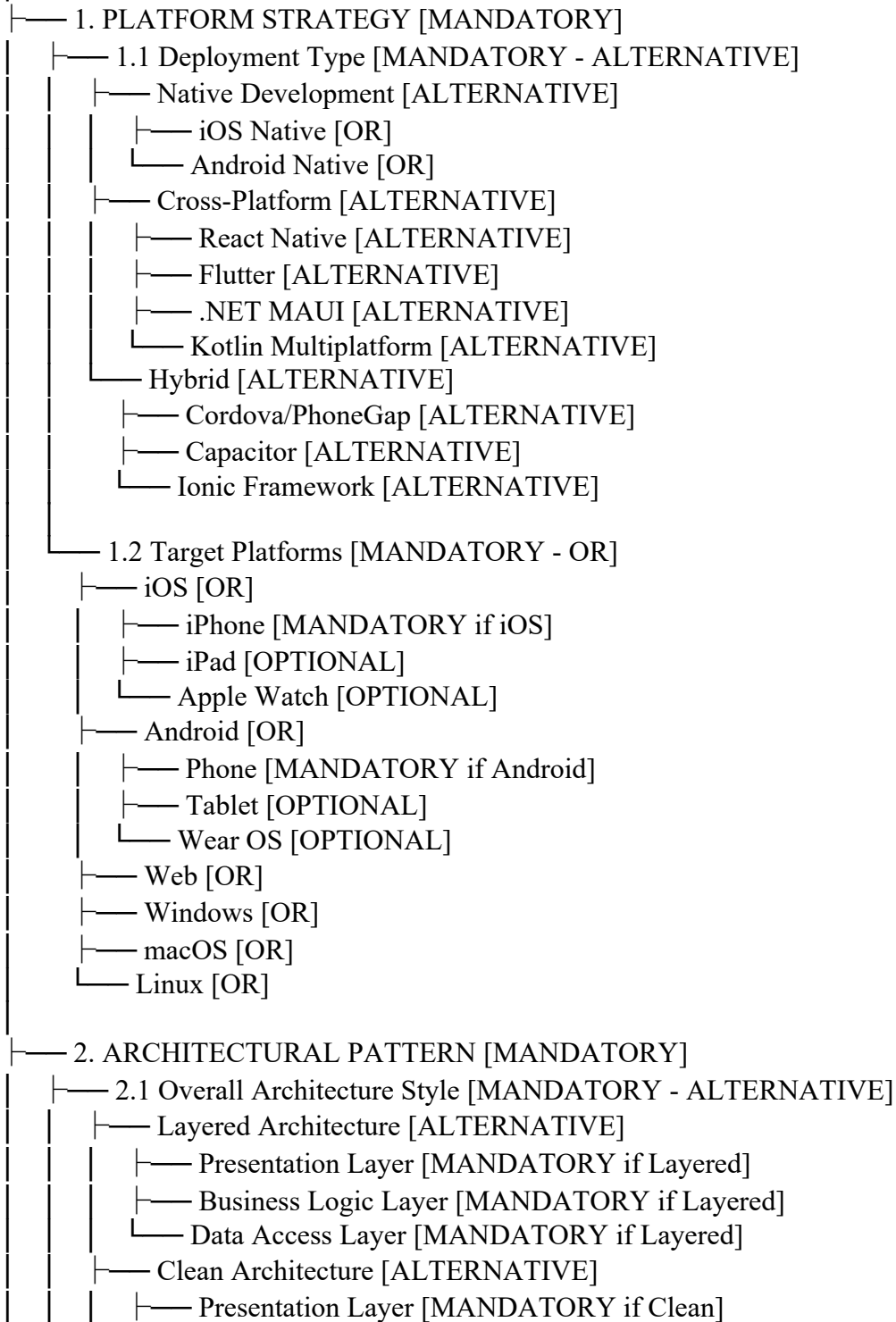


### 3. Constraints principales

- IF (Deployment Type = Flutter) THEN Target Platform ∈ {Android, iOS}
- IF (Deployment Type = Android Native) THEN Target Platform = Android
- IF (Deployment Type = iOS Native) THEN Target Platform = iOS
- IF (Deployment Type = Flutter) THEN UI Rendering Approach = Cross-Platform Widgets AND HTTP Client = Dio
- IF (Deployment Type = iOS Native) THEN UI Rendering Approach = Native Declarative AND HTTP Client = Native HTTP Client
- IF (Push Notifications = TRUE) THEN (Push Provider = TRUE) AND (exactly-one(FirebaseCloudMessaging, AirShip, OneSignal))
- IF (Data Source = Local-Only) THEN API Architecture Style = EXCLUDED AND HTTP Client = EXCLUDED
- IF (API Architecture Style = REST) AND (Deployment Type ∈ {Android Native, iOS Native}) THEN HTTP Client = Native HTTP Client
- IF (API Architecture Style = GraphQL) AND (Deployment Type = Flutter) THEN HTTP Client = Dio
- IF (API Architecture Style = GraphQL) AND (Deployment Type ∈ {Android Native, iOS Native}) THEN HTTP Client = Native HTTP Client
- IF (Hive = TRUE) THEN Deployment Type = Flutter
- IF (Deployment Type) != Flutter THEN Hive = FALSE
- IF (SQLite = TRUE) THEN Deployment Type ∈ {Android Native, iOS Native}
- IF (Presentation Pattern = MVI) THEN Overall Style = Clean Architecture
- IF (Domain-Specific Extension = Financial Features) AND (Complejidad = Alta) THEN Biometric = TRUE AND Data Encryption = TRUE

## 10.4 MOBILE APPLICATION ARCHITECTURE FEATURE MODEL (MAAFM)

### Mobile Application Architecture [ROOT - MANDATORY]



- └─ Domain Layer [MANDATORY if Clean]
  - └─ Use Cases/Interactors [MANDATORY if Domain Layer]
  - └─ Domain Entities [MANDATORY if Domain Layer]
  - └─ Repository Interfaces [MANDATORY if Domain Layer]
- └─ Data Layer [MANDATORY if Clean]
  - └─ Repository Implementation [MANDATORY if Data Layer]
  - └─ Data Sources [MANDATORY if Data Layer]
  - └─ Data Models/DTOs [MANDATORY if Data Layer]
- └─ Dependency Inversion [MANDATORY if Clean]
- └─ Hexagonal Architecture [ALTERNATIVE]
  - └─ Core Domain [MANDATORY if Hexagonal]
  - └─ Ports [MANDATORY if Hexagonal]
  - └─ Adapters [MANDATORY if Hexagonal]
- └─ Onion Architecture [ALTERNATIVE]
  - └─ Domain Model (Center) [MANDATORY if Onion]
  - └─ Domain Services [MANDATORY if Onion]
  - └─ Application Services [MANDATORY if Onion]
  - └─ Infrastructure (Outer) [MANDATORY if Onion]

## └─ 2.2 Modularization Strategy [MANDATORY - ALTERNATIVE]

- └─ Monolithic [ALTERNATIVE]
  - └─ Single Deployable Unit [MANDATORY if Monolithic]
  - └─ Internal Package Structure [OPTIONAL]
- └─ Modular (Multi-Module) [ALTERNATIVE]
  - └─ Feature Modules [OPTIONAL]
  - └─ Core Module [MANDATORY if Modular]
  - └─ Shared Module [OPTIONAL]
  - └─ Dynamic Feature Modules [OPTIONAL]
    - └─ On-Demand Delivery [OPTIONAL]
    - └─ Conditional Delivery [OPTIONAL]
- └─ Micro-Apps (Multi-Repo) [ALTERNATIVE]
  - └─ Independent App Module [MANDATORY if Micro-Apps]
  - └─ Core Module [MANDATORY if Micro-Apps]
  - └─ Shared Common Modules [MANDATORY if Micro-Apps]
  - └─ Inter-App Communication [MANDATORY if Micro-Apps]
    - └─ Deep Links [OR]
    - └─ URL Schemes [OR]
    - └─ Shared Container/Storage [OR]
  - └─ Dynamic Feature Modules [OPTIONAL - ALTERNATIVE if selected]
    - └─ On-Demand Delivery [ALTERNATIVE]
    - └─ Conditional Delivery [ALTERNATIVE]
    - └─ Install-Time Delivery [ALTERNATIVE]

└─ 2.3 Presentation Pattern [MANDATORY - ALTERNATIVE]

└─ MVC (Model-View-Controller) [ALTERNATIVE]

└─ MVP (Model-View-Presenter) [ALTERNATIVE]

└─ MVVM (Model-View-ViewModel) [ALTERNATIVE]

└─ Data Binding [OPTIONAL]

└─ Reactive Extensions [OPTIONAL]

└─ Observable State [MANDATORY if MVVM]

└─ MVI (Model-View-Intent) [ALTERNATIVE]

└─ Unidirectional Data Flow [MANDATORY if MVI]

└─ Immutable State [MANDATORY if MVI]

└─ VIPER [ALTERNATIVE]

└─ View [MANDATORY if VIPER]

└─ Interactor [MANDATORY if VIPER]

└─ Presenter [MANDATORY if VIPER]

└─ Entity [MANDATORY if VIPER]

└─ Router [MANDATORY if VIPER]

└─ 2.4 Code Organization [MANDATORY - ALTERNATIVE]

└─ Package by Feature [ALTERNATIVE]

└─ Package by Layer [ALTERNATIVE]

└─ Package by Domain [ALTERNATIVE]

└─ 3. DATA MANAGEMENT LAYER [MANDATORY]

└─ 3.1 Data Source Strategy [MANDATORY - ALTERNATIVE]

└─ Local-Only [ALTERNATIVE]

└─ Remote-Only [ALTERNATIVE]

└─ Hybrid (Local + Remote) [ALTERNATIVE]

└─ 3.2 Local Storage [MANDATORY if (Data Source ≠ Remote-Only)]

└─ Structured Storage [OR]

└─ Relational Database [OPTIONAL]

└─ SQLite [ALTERNATIVE]

└─ Realm [ALTERNATIVE]

└─ CoreData [ALTERNATIVE]

└─ Encryption [OPTIONAL]

└─ NoSQL Database [OPTIONAL]

└─ Document Store [OR]

└─ Key-Value Store [OR]

└─ Object Database [OR]

└─ Preferences Storage [OR]

└─ Key-Value Preferences [MANDATORY if Preferences]

└─ File System Storage [OR]

└─ Internal Storage [OR]

└─ External Storage [OR]

└─ Cache Directory [OR]

- └─ Documents Directory [OR]
- └─ In-Memory Storage [OR]
- └─ LRU Cache [OR]
- └─ Memory Cache [OR]
- └─ Session Storage [OR]

### └─ 3.3 Data Synchronization [MANDATORY if (Data Source = Hybrid)]

- └─ Sync Strategy [MANDATORY if Data Sync - ALTERNATIVE]
  - └─ Real-time Sync [ALTERNATIVE]
  - └─ Periodic Sync [ALTERNATIVE]
  - └─ On-demand Sync [ALTERNATIVE]
  - └─ Event-driven Sync [ALTERNATIVE]
- └─ Conflict Resolution [MANDATORY if Data Sync - ALTERNATIVE]
  - └─ Last Write Wins [ALTERNATIVE]
  - └─ First Write Wins [ALTERNATIVE]
  - └─ Manual Resolution [ALTERNATIVE]
  - └─ Merge Strategy [ALTERNATIVE]
- └─ Offline-First Strategy [OPTIONAL]
  - └─ Local-first Persistence [MANDATORY if Offline-First]
  - └─ Operation Queue [MANDATORY if Offline-First]
  - └─ Background Sync [OPTIONAL]

### └─ 3.4 Caching Strategy [MANDATORY]

- └─ Cache Policy [MANDATORY - ALTERNATIVE]
  - └─ Cache-First [ALTERNATIVE]
  - └─ Network-First [ALTERNATIVE]
  - └─ Cache-Then-Network [ALTERNATIVE]
  - └─ Network-Only [ALTERNATIVE]
  - └─ Cache-Only [ALTERNATIVE]
- └─ Cache Invalidation [MANDATORY - OR]
  - └─ TTL (Time To Live) [OR]
  - └─ Manual Invalidation [OR]
  - └─ Event-Based [OR]
  - └─ LRU Eviction [OR]
- └─ Image Caching [OPTIONAL]
- └─ API Response Caching [OPTIONAL]

### └─ 3.5 State Management [MANDATORY]

- └─ State Scope [MANDATORY - OR]
  - └─ Local Component State [OR]
  - └─ Shared State [OR]
  - └─ Global Application State [OR]
- └─ State Management Approach [MANDATORY - ALTERNATIVE]
  - └─ Centralized Store [ALTERNATIVE]
  - └─ Distributed State [ALTERNATIVE]

- └─ Reactive Streams [ALTERNATIVE]
  - └─ 3.6 Data Repository Pattern [MANDATORY]
    - └─ Single Source of Truth [MANDATORY]
    - └─ Data Mapper [OPTIONAL]
    - └─ DTO Pattern [OPTIONAL]
    - └─ Data Source Abstraction [MANDATORY]
- └─ 4. API LAYER & NETWORKING [MANDATORY if (Data Source ≠ Local-Only)]
  - └─ 4.1 API Architecture Style [MANDATORY if Networking - ALTERNATIVE]
    - └─ REST [ALTERNATIVE]
      - └─ HTTP Methods [MANDATORY if REST]
      - └─ Resource-based URIs [MANDATORY if REST]
      - └─ JSON Parsing [MANDATORY]
      - └─ HATEOAS [OPTIONAL]
      - └─ XML Support [OPTIONAL]
      - └─ Multipart/Form-Data [OPTIONAL]
    - └─ GraphQL [ALTERNATIVE]
      - └─ Query [MANDATORY if GraphQL]
      - └─ Mutation [MANDATORY if GraphQL]
      - └─ Subscription [OPTIONAL]
      - └─ Schema Introspection [OPTIONAL]
    - └─ gRPC [ALTERNATIVE]
      - └─ Protocol Buffers [MANDATORY if gRPC]
      - └─ Bidirectional Streaming [OPTIONAL]
      - └─ HTTP/2 [MANDATORY if gRPC]
    - └─ WebSocket [ALTERNATIVE]
      - └─ Full-Duplex Communication [MANDATORY if WebSocket]
      - └─ Event-driven Messaging [MANDATORY if WebSocket]
  - └─ 4.2 HTTP Client Implementation [MANDATORY if (API Style = REST OR GraphQL)]
    - └─ Native HTTP Client [ALTERNATIVE]
    - └─ Third-Party HTTP Library [ALTERNATIVE]
  - └─ 4.3 Backend Integration [OPTIONAL]
    - └─ Backend-as-a-Service (BaaS) [OPTIONAL - ALTERNATIVE]
      - └─ Firebase [ALTERNATIVE]
        - └─ Firestore [OR]
        - └─ Realtime Database [OR]
        - └─ Cloud Functions [OR]
        - └─ Cloud Storage [OR]
      - └─ AWS Amplify [ALTERNATIVE]
      - └─ Azure Mobile Apps [ALTERNATIVE]
      - └─ Supabase [ALTERNATIVE]
      - └─ Parse Server [ALTERNATIVE]

- └─ Custom Backend [ALTERNATIVE]

- └─ 4.4 Network Security [MANDATORY if Networking]

- └─ Transport Security [MANDATORY if Networking]

- └─ TLS/SSL 1.3 [MANDATORY if Transport Security]

- └─ Certificate Pinning [OPTIONAL]

- └─ Public Key Pinning [OPTIONAL]

- └─ Certificate Validation [MANDATORY if Transport Security]

- └─ API Security [MANDATORY if Networking]

- └─ API Key Management [MANDATORY if API Security]

- └─ Request Signing [OPTIONAL]

- └─ Token-based Auth [OPTIONAL]

- └─ Network Security Policy [OPTIONAL]

- └─ 4.5 Request/Response Handling [MANDATORY if Networking]

- └─ Interceptors [OPTIONAL]

- └─ Authentication Interceptor [OR]

- └─ Logging Interceptor [OR]

- └─ Header Interceptor [OR]

- └─ Retry Interceptor [OR]

- └─ Serialization [MANDATORY if Networking - OR]

- └─ JSON [OR]

- └─ Protocol Buffers [OR]

- └─ XML [OR]

- └─ MessagePack [OR]

- └─ Error Handling [MANDATORY if Networking]

- └─ Network Errors [MANDATORY if Error Handling]

- └─ HTTP Status Codes [MANDATORY if Error Handling]

- └─ Timeout Handling [MANDATORY if Error Handling]

- └─ Retry Logic [OPTIONAL]

- └─ Exponential Backoff [ALTERNATIVE]

- └─ Fixed Retry [ALTERNATIVE]

- └─ Circuit Breaker [ALTERNATIVE]

- └─ Request Rate Limiting [OPTIONAL]

- └─ 4.6 Connectivity Management [MANDATORY if Networking]

- └─ Network State Monitoring [MANDATORY if Connectivity]

- └─ Connection Type Detection [MANDATORY if State Monitoring]

- └─ WiFi [OR]

- └─ Cellular (3G/4G/5G) [OR]

- └─ No Connection [MANDATORY if Connection Detection]

- └─ Reachability Check [MANDATORY if State Monitoring]

- └─ Adaptive Networking [OPTIONAL]

- └─ Bandwidth Optimization [OR]

- └─ Quality Adjustment [OR]

- └─ Data Saver Mode [OR]
- └─ Offline Request Queue [OPTIONAL]

#### 4.7 API Contract & Versioning [MANDATORY if Networking]

- └─ Versioning Strategy [MANDATORY if API Contract - ALTERNATIVE]
  - └─ URI Versioning [ALTERNATIVE]
  - └─ Header Versioning [ALTERNATIVE]
  - └─ Media Type Versioning [ALTERNATIVE]
  - └─ Query Parameter Versioning [ALTERNATIVE]
- └─ API Documentation [OPTIONAL]
  - └─ OpenAPI/Swagger [ALTERNATIVE]
  - └─ GraphQL Schema [ALTERNATIVE]
- └─ Contract Testing [OPTIONAL]
- └─ Deprecation Handling [OPTIONAL]

### 5. SECURITY ARCHITECTURE [MANDATORY]

#### 5.1 Authentication [MANDATORY - OR]

- └─ Local Authentication [OR]
  - └─ PIN/Password [OR]
  - └─ Pattern Lock [OR]
  - └─ Biometric [OR]
    - └─ Fingerprint [OR]
    - └─ Face Recognition [OR]
- └─ Remote Authentication [OR]
  - └─ Token-Based [ALTERNATIVE]
    - └─ JWT [ALTERNATIVE]
    - └─ OAuth 2.0 [ALTERNATIVE]
    - └─ OpenID Connect [ALTERNATIVE]
    - └─ SAML [ALTERNATIVE]
  - └─ Session-Based [ALTERNATIVE]
- └─ Multi-Factor Authentication (MFA) [OR]
  - └─ SMS OTP [OR]
  - └─ Email OTP [OR]
  - └─ TOTP (Authenticator App) [OR]
  - └─ Push Notification [OR]
  - └─ Hardware Token [OR]
- └─ Federated Authentication [OR]
  - └─ Social Authentication [OPTIONAL]
    - └─ Google Sign-In [OR]
    - └─ Apple Sign-In [OR]
    - └─ Facebook Login [OR]
    - └─ Microsoft Account [OR]
  - └─ Enterprise SSO [OPTIONAL]
    - └─ SAML 2.0 [ALTERNATIVE]
    - └─ Azure AD [ALTERNATIVE]

5.2 Authorization [MANDATORY]

Access Control Model [MANDATORY - ALTERNATIVE]

RBAC (Role-Based) [ALTERNATIVE]

ABAC (Attribute-Based) [ALTERNATIVE]

PBAC (Policy-Based) [ALTERNATIVE]

ACL (Access Control List) [ALTERNATIVE]

Token Management [MANDATORY if Token-Based Auth]

Token Storage [MANDATORY if Token Management]

Token Refresh [MANDATORY if Token Management]

Token Revocation [OPTIONAL]

5.3 Data Protection [MANDATORY]

Data at Rest Encryption [MANDATORY]

Database Encryption [MANDATORY if Data at Rest]

File Encryption [OPTIONAL]

Full Disk Encryption [OPTIONAL]

Data in Transit Encryption [MANDATORY]

TLS/SSL [MANDATORY if Data in Transit]

End-to-End Encryption [OPTIONAL]

Message Encryption [OPTIONAL]

Secure Key Storage [MANDATORY]

Hardware-backed Keystore [MANDATORY if Secure Key Storage]

Encryption Algorithms [MANDATORY - OR]

AES (Symmetric) [OR]

RSA (Asymmetric) [OR]

Elliptic Curve [OR]

5.4 Application Security [MANDATORY]

Code Protection [OPTIONAL]

Code Obfuscation [OR]

Anti-Tampering [OR]

Anti-Debugging [OR]

Runtime Protection [OPTIONAL]

Root/Jailbreak Detection [OR]

Emulator Detection [OR]

Integrity Verification [OR]

Secure Coding Practices [MANDATORY]

Input Validation [MANDATORY if Secure Coding]

Output Encoding [MANDATORY if Secure Coding]

Secure Data Handling [MANDATORY if Secure Coding]

5.5 Privacy & Compliance [MANDATORY]

Privacy Framework [MANDATORY]

Privacy Policy [MANDATORY if Privacy Framework]

- └─ Terms of Service [MANDATORY if Privacy Framework]
- └─ User Consent Management [MANDATORY if Privacy Framework]
- └─ Data Minimization [MANDATORY if Privacy Framework]
- └─ Regulatory Compliance [OPTIONAL - OR]
  - └─ GDPR [OR]
  - └─ CCPA [OR]
  - └─ HIPAA [OR]
  - └─ PCI DSS [OR]
  - └─ SOC 2 [OR]
  - └─ ISO 27001 [OR]
- └─ Data Subject Rights [OPTIONAL]
  - └─ Right to Access [OR]
  - └─ Right to Deletion [OR]
  - └─ Right to Portability [OR]
  - └─ Right to Rectification [OR]
- └─ 5.6 Security Monitoring [OPTIONAL]
  - └─ Threat Detection [OR]
  - └─ Security Logging [OR]
  - └─ Vulnerability Scanning [OR]
  - └─ Security Incident Response [OR]

## 6. USER INTERFACE LAYER [MANDATORY]

- └─ 6.1 UI Rendering Approach [MANDATORY - ALTERNATIVE]
  - └─ Native Declarative [ALTERNATIVE]
  - └─ Native Imperative [ALTERNATIVE]
  - └─ Cross-Platform Widgets [ALTERNATIVE]
  - └─ Web-Based UI [ALTERNATIVE]
    - └─ WebView Container [MANDATORY if Web-Based UI]
- └─ 6.2 Navigation Architecture [MANDATORY]
  - └─ Navigation Pattern [MANDATORY - OR]
    - └─ Stack Navigation [OR]
    - └─ Tab Navigation [OR]
    - └─ Drawer Navigation [OR]
    - └─ Modal Navigation [OR]
    - └─ Bottom Navigation [OR]
  - └─ Deep Linking [OPTIONAL]
    - └─ Universal Links [OR]
    - └─ App Links [OR]
    - └─ Custom URL Schemes [OR]
- └─ 6.3 UI State Management [MANDATORY]
  - └─ Local State [OPTIONAL]
  - └─ Global State [OPTIONAL]

- └─ View State [MANDATORY - OR]
  - └─ Loading State [MANDATORY if View State]
  - └─ Success State [MANDATORY if View State]
  - └─ Error State [MANDATORY if View State]
  - └─ Empty State [MANDATORY if View State]
  - └─ Placeholder/Skeleton [OPTIONAL]
- └─ Lifecycle Management [MANDATORY]
  - └─ Creation [MANDATORY if Lifecycle]
  - └─ Resume/Pause [MANDATORY if Lifecycle]
  - └─ Destruction [MANDATORY if Lifecycle]
  - └─ State Restoration [OPTIONAL]

#### └─ 6.4 Responsive Design [MANDATORY]

- └─ Adaptive Layouts [MANDATORY]
  - └─ Phone Layout [MANDATORY if Adaptive]
  - └─ Tablet Layout [OPTIONAL]
  - └─ Foldable Support [OPTIONAL]
  - └─ Desktop Layout [OPTIONAL]
- └─ Orientation Support [MANDATORY]
  - └─ Portrait [MANDATORY if Orientation]
  - └─ Landscape [OPTIONAL]
- └─ Screen Density Support [MANDATORY]
- └─ Multi-Window Support [OPTIONAL]
  - └─ Split Screen [OR]
  - └─ Picture-in-Picture [OR]

#### └─ 6.5 Accessibility (a11y) [MANDATORY]

- └─ Screen Reader Support [MANDATORY]
- └─ Semantic Labeling [MANDATORY]
- └─ Content Descriptions [MANDATORY]
- └─ Dynamic Text Sizing [MANDATORY]
- └─ Color Contrast [MANDATORY]
- └─ Touch Target Size [MANDATORY]
- └─ Focus Management [OPTIONAL]
- └─ Keyboard Navigation [OPTIONAL]
- └─ Accessibility Standards [OPTIONAL]
  - └─ WCAG 2.1 AA [ALTERNATIVE]
  - └─ WCAG 2.1 AAA [ALTERNATIVE]

#### └─ 6.6 Theming & Styling [MANDATORY]

- └─ Theme Support [MANDATORY]
  - └─ Light Theme [MANDATORY if Theme Support]
  - └─ Dark Theme [OPTIONAL]
  - └─ System Theme [OPTIONAL]
  - └─ High Contrast Theme [OPTIONAL]

- └─ Design System [OPTIONAL]
  - └─ Material Design [ALTERNATIVE]
  - └─ Human Interface Guidelines [ALTERNATIVE]
  - └─ Fluent Design [ALTERNATIVE]
  - └─ Custom Design System [ALTERNATIVE]
- └─ Typography System [MANDATORY]
- └─ Color System [MANDATORY]
- └─ Spacing System [MANDATORY]

#### └─ 6.7 Animations & Transitions [OPTIONAL]

- └─ View Animations [OR]
- └─ Transition Animations [OR]
- └─ Gesture Animations [OR]

#### └─ 6.8 UI Components Library [MANDATORY]

- └─ Form Components [MANDATORY]
- └─ Display Components [MANDATORY]
- └─ List Components [OR]
  - └─ Flat Lists [OR]
  - └─ Grid Lists [OR]
  - └─ Infinite Scroll [OR]
- └─ Navigation Components [OR]
  - └─ Tab Bars [OR]
  - └─ Navigation Bars [OR]
  - └─ Bottom Sheets [OR]
- └─ Feedback Components [OR]
  - └─ Dialogs/Alerts [OR]
  - └─ Toasts/Snackbars [OR]
  - └─ Progress Indicators [OR]
  - └─ Pull-to-Refresh [OR]

### └─ 7. DEVICE CAPABILITIES & INTEGRATIONS [OPTIONAL]

#### └─ 7.1 Sensors & Hardware [OPTIONAL - OR]

- └─ Camera [OR]
  - └─ Photo Capture [OR]
  - └─ Video Recording [OR]
  - └─ QR/Barcode Scanner [OR]
  - └─ OCR (Text Recognition) [OR]
- └─ Location Services [OR]
  - └─ GPS [OR]
  - └─ Geofencing [OR]
  - └─ Geocoding [OR]
  - └─ Location Tracking [OR]
- └─ Motion Sensors [OR]
  - └─ Accelerometer [OR]

- └─ Gyroscope [OR]
- └─ Magnetometer [OR]
- └─ Step Counter [OR]
- └─ Environmental Sensors [OR]
  - └─ Proximity Sensor [OR]
  - └─ Light Sensor [OR]
  - └─ Barometer [OR]
  - └─ Temperature Sensor [OR]
- └─ Audio [OR]
  - └─ Microphone [OR]
  - └─ Audio Recording [OR]
  - └─ Speech Recognition [OR]
  - └─ Voice Commands [OR]
- └─ Connectivity Hardware [OR]
  - └─ Bluetooth [OR]
    - └─ Classic Bluetooth [OR]
    - └─ BLE (Low Energy) [OR]
    - └─ Beacon Support [OR]
  - └─ NFC [OR]
    - └─ NFC Reading [OR]
    - └─ NFC Writing [OR]
    - └─ NFC Payments [OR]
  - └─ WiFi Direct [OR]
- └─ Haptics [OR]
  - └─ Vibration [OR]
  - └─ Haptic Feedback [OR]

## └─ 7.2 Augmented Reality (AR) [OPTIONAL]

- └─ AR Framework [MANDATORY if AR]
- └─ 3D Object Rendering [OR]
- └─ Plane Detection [OR]
- └─ Image Tracking [OR]
- └─ Face Tracking [OR]

## └─ 7.3 Media Management [OPTIONAL]

- └─ Gallery Access [OPTIONAL]
  - └─ Photo Library [OR]
  - └─ Video Library [OR]
- └─ Image Processing [OPTIONAL]
  - └─ Filters [OR]
  - └─ Cropping [OR]
  - └─ Resizing [OR]
  - └─ Compression [OR]
- └─ Media Playback [OPTIONAL]
  - └─ Audio Player [OR]

- └─ Video Player [OR]
- └─ Streaming [OR]

#### └─ 7.4 OS Integration [OPTIONAL]

- └─ Contacts [OR]
- └─ Calendar [OR]
- └─ Phone/Dialer [OR]
- └─ SMS/MMS [OR]
- └─ Email [OR]
- └─ Share Sheet [OR]
- └─ App Shortcuts [OR]
- └─ Home Screen Widgets [OR]

#### └─ 7.5 Maps & Location Services [OPTIONAL]

- └─ Map Display [MANDATORY if Maps]
- └─ Markers & Annotations [OR]
- └─ Route Planning [OR]
- └─ Turn-by-Turn Navigation [OR]
- └─ Places API [OR]

#### └─ 7.6 Payment Integration [OPTIONAL]

- └─ In-App Purchases [OR]
- └─ Payment Gateway [OR]
- └─ Mobile Wallets [OR]
  - └─ Apple Pay [OR]
  - └─ Google Pay [OR]
  - └─ Samsung Pay [OR]
- └─ Subscription Management [OR]

#### └─ 7.7 Document Management [OPTIONAL]

- └─ PDF Operations [OPTIONAL]
  - └─ PDF Viewer [OR]
  - └─ PDF Generator [OR]
- └─ Document Scanner [OR]
- └─ File Picker [OR]
- └─ File Sharing [OR]

### └─ 8. REAL-TIME COMMUNICATION [OPTIONAL]

#### └─ 8.1 Push Notifications [OPTIONAL]

- └─ Remote Notifications [OPTIONAL]
  - └─ Push Provider [MANDATORY if Remote Notifications - ALTERNATIVE]
    - └─ Firebase Cloud Messaging [ALTERNATIVE]
    - └─ Apple Push Notification Service [ALTERNATIVE]
    - └─ OneSignal [ALTERNATIVE]
    - └─ Amazon SNS [ALTERNATIVE]

- └─ Notification Channels [OPTIONAL]
  - └─ Local Notifications [OPTIONAL]
    - └─ Scheduled Notifications [OR]
  - └─ Rich Notifications [OPTIONAL]
    - └─ Images [OR]
    - └─ Actions [OR]
    - └─ Custom UI [OR]
  - └─ Notification Management [OPTIONAL]
    - └─ Notification Grouping [OR]
    - └─ Notification Badges [OR]
- └─ 8.2 Bidirectional Messaging [OPTIONAL]
  - └─ Messaging Protocol [MANDATORY if Bidirectional - ALTERNATIVE]
    - └─ WebSocket [ALTERNATIVE]
    - └─ Server-Sent Events [ALTERNATIVE]
    - └─ Long Polling [ALTERNATIVE]
    - └─ MQTT [ALTERNATIVE]
  - └─ Message Queue [OR]
  - └─ Presence Detection [OR]
- └─ 8.3 Video/Audio Calling [OPTIONAL]
  - └─ WebRTC [MANDATORY if Video/Audio Calling]
  - └─ Voice Calling [OR]
  - └─ Video Calling [OR]
  - └─ Conference Calling [OR]
  - └─ Screen Sharing [OR]
- └─ 8.4 Chat/Messaging [OPTIONAL]
  - └─ Text Messaging [MANDATORY if Chat]
  - └─ Media Sharing [OR]
  - └─ Typing Indicators [OR]
  - └─ Read Receipts [OR]
  - └─ Message Threading [OR]
- └─ 8.5 Email Integration [OPTIONAL]
  - └─ Email Composition [OR]
  - └─ Email Sending [OR]
- └─ 9. BACKGROUND PROCESSING [OPTIONAL]
  - └─ 9.1 Background Execution [OPTIONAL]
    - └─ Background Services [OR]
    - └─ Foreground Services [OR]
    - └─ Background Fetch [OR]
  - └─ 9.2 Work Scheduling [OPTIONAL]

- └─ Periodic Work [OR]
- └─ One-Time Work [OR]
- └─ Deferred Work [OR]
- └─ Expedited Work [OR]

### └─ 9.3 Background Sync [OPTIONAL]

- └─ Data Synchronization [OR]
- └─ Content Download [OR]
- └─ Upload Queue [OR]

### └─ 9.4 Background Constraints [OPTIONAL]

- └─ Network Required [OR]
- └─ Battery Level [OR]
- └─ Device Idle [OR]
- └─ Storage Space [OR]

## └─ 10. PERFORMANCE OPTIMIZATION [MANDATORY]

### └─ 10.1 Memory Management [MANDATORY]

- └─ Memory Leak Prevention [MANDATORY]
- └─ Object Pooling [OPTIONAL]
- └─ Weak References [OPTIONAL]
- └─ Memory Profiling [OPTIONAL]

### └─ 10.2 Battery Optimization [MANDATORY]

- └─ Doze Mode Handling [OPTIONAL]
- └─ Battery Saver Mode [OPTIONAL]
- └─ Wake Lock Management [OPTIONAL]
- └─ Background Execution Limits [OPTIONAL]

### └─ 10.3 Network Optimization [MANDATORY if Networking]

- └─ Request Batching [OPTIONAL]
- └─ Request Deduplication [OPTIONAL]
- └─ Response Compression [OPTIONAL]
- └─ Image Optimization [OPTIONAL]
- └─ Prefetching [OPTIONAL]

### └─ 10.4 Rendering Performance [MANDATORY]

- └─ View Recycling [MANDATORY if UI Layer]
- └─ Lazy Loading [OPTIONAL]
- └─ Pagination [OPTIONAL]
- └─ Virtual Scrolling [OPTIONAL]
- └─ GPU Acceleration [OPTIONAL]
- └─ Frame Rate Target [MANDATORY]
  - └─ 60 FPS [ALTERNATIVE]
  - └─ 120 FPS [ALTERNATIVE]

- └─ Overdraw Reduction [OPTIONAL]

- └─ 10.5 App Size Optimization [MANDATORY]

- └─ Code Shrinking [MANDATORY]

- └─ Dead Code Elimination [MANDATORY if Code Shrinking]

- └─ Unused Resource Removal [MANDATORY if Code Shrinking]

- └─ Resource Optimization [OPTIONAL]

- └─ Image Compression [OR]

- └─ Vector Graphics [OR]

- └─ WebP Format [OR]

- └─ Asset Delivery [OPTIONAL]

- └─ On-Demand Assets [OR]

- └─ Asset Packs [OR]

- └─ Native Library Optimization [OPTIONAL]

- └─ ABI Splitting [OR]

- └─ 10.6 Startup Performance [MANDATORY]

- └─ Launch Time Optimization [MANDATORY]

- └─ Cold Start Optimization [MANDATORY if Launch Time]

- └─ Warm Start Optimization [OPTIONAL]

- └─ Hot Start Optimization [OPTIONAL]

- └─ Lazy Initialization [OPTIONAL]

- └─ Splash Screen Strategy [OPTIONAL]

- └─ Critical Path Optimization [MANDATORY]

- └─ 11. TESTING STRATEGY [MANDATORY]

- └─ 11.1 Unit Testing [OPTIONAL]

- └─ Test Framework [MANDATORY if Unit Testing]

- └─ Test Pattern [MANDATORY if Unit Testing - ALTERNATIVE]

- └─ Arrange-Act-Assert [ALTERNATIVE]

- └─ Given-When-Then [ALTERNATIVE]

- └─ Test Doubles [OPTIONAL - OR]

- └─ Mocks [OR]

- └─ Stubs [OR]

- └─ Fakes [OR]

- └─ Spies [OR]

- └─ Code Coverage [OPTIONAL]

- └─ Coverage Target [OPTIONAL]

- └─ Coverage Reports [OPTIONAL]

- └─ 11.2 Integration Testing [OPTIONAL]

- └─ API Integration Tests [OR]

- └─ Database Integration Tests [OR]

- └─ Component Integration Tests [OR]

- └─ Contract Testing [OR]

└─ 11.3 UI Testing [OPTIONAL]

- └─ UI Test Framework [MANDATORY if UI Testing]
- └─ End-to-End Tests [OR]
- └─ Snapshot Testing [OR]
- └─ Visual Regression Testing [OR]

└─ 11.4 Performance Testing [OPTIONAL]

- └─ Load Testing [OR]
- └─ Stress Testing [OR]
- └─ CPU Profiling [OR]
- └─ Memory Profiling [OR]
- └─ Battery Profiling [OR]

└─ 11.5 Security Testing [OPTIONAL]

- └─ Static Application Security Testing (SAST) [OR]
- └─ Dynamic Application Security Testing (DAST) [OR]
- └─ Penetration Testing [OR]
- └─ Dependency Vulnerability Scanning [OR]

└─ 11.6 Test Automation & CI [OPTIONAL]

- └─ Automated Test Execution [OR]
- └─ Test Parallelization [OR]
- └─ Test Reporting [OR]

└─ 12. OBSERVABILITY & MONITORING [MANDATORY]

└─ 12.1 Crash Reporting [MANDATORY]

- └─ Crash Detection [MANDATORY if Crash Reporting]
- └─ Crash Symbolication [MANDATORY if Crash Reporting]
- └─ Crash Grouping [OPTIONAL]
- └─ Crash Alerts [OPTIONAL]

└─ 12.2 Analytics [MANDATORY]

- └─ Event Tracking [MANDATORY if Analytics]
  - └─ Screen Views [MANDATORY if Event Tracking]
  - └─ User Actions [MANDATORY if Event Tracking]
  - └─ Custom Events [OPTIONAL]
  - └─ Conversion Events [OPTIONAL]
- └─ User Analytics [OPTIONAL]
  - └─ User Properties [OR]
  - └─ User Segmentation [OR]
  - └─ User Journey [OR]
  - └─ Cohort Analysis [OR]
- └─ Funnel Analysis [OPTIONAL]
- └─ Retention Analysis [OPTIONAL]

- └─ Attribution [OPTIONAL]
  - └─ Marketing Attribution [OR]
  - └─ Deep Link Attribution [OR]

### └─ 12.3 Performance Monitoring [MANDATORY]

- └─ Application Performance [OPTIONAL]
  - └─ App Start Time [OR]
  - └─ Screen Rendering Time [OR]
  - └─ Frame Rate Monitoring [OR]
  - └─ Memory Usage [OR]
- └─ Network Performance [MANDATORY if Networking]
  - └─ HTTP Request Tracking [MANDATORY if Network Performance]
  - └─ Response Time [OPTIONAL]
  - └─ Success/Failure Rate [OPTIONAL]
  - └─ Payload Size [OPTIONAL]
- └─ Custom Metrics [OPTIONAL]

### └─ 12.4 Logging [MANDATORY]

- └─ Log Levels [MANDATORY if Logging]
  - └─ Verbose/Debug [MANDATORY if Log Levels]
  - └─ Info [MANDATORY if Log Levels]
  - └─ Warning [MANDATORY if Log Levels]
  - └─ Error [MANDATORY if Log Levels]
  - └─ Fatal/Critical [MANDATORY if Log Levels]
- └─ Structured Logging [OPTIONAL]
- └─ Log Aggregation [OPTIONAL]
- └─ Remote Logging [OPTIONAL]
- └─ Log Sanitization [MANDATORY]
  - └─ PII Removal [MANDATORY if Log Sanitization]
  - └─ Sensitive Data Masking [MANDATORY if Log Sanitization]

### └─ 12.5 Real User Monitoring (RUM) [OPTIONAL]

- └─ Session Recording [OR]
- └─ User Journey Tracking [OR]
- └─ Behavior Analytics [OR]
- └─ Heatmaps [OR]

### └─ 12.6 User Feedback [OPTIONAL]

- └─ In-App Feedback [OR]
- └─ Bug Reporting [OR]
- └─ Feature Requests [OR]
- └─ Rating Prompts [OR]
- └─ App Store Review Monitoring [OR]

## └─ 13. CI/CD & DEVOPS [MANDATORY]

└─ 13.1 Version Control [MANDATORY]

- └─ Git [MANDATORY]
- └─ Branching Strategy [MANDATORY - ALTERNATIVE]
  - └─ Git Flow [ALTERNATIVE]
  - └─ GitHub Flow [ALTERNATIVE]
  - └─ Trunk-Based Development [ALTERNATIVE]
  - └─ GitLab Flow [ALTERNATIVE]
- └─ Code Review [OPTIONAL]
  - └─ Pull Request Reviews [OR]
  - └─ Automated Code Review [OR]

└─ 13.2 Build System [MANDATORY]

- └─ Build Configuration [MANDATORY]
  - └─ Build Environments [MANDATORY if Build Configuration]
    - └─ Development [MANDATORY if Build Environments]
    - └─ Staging [OPTIONAL]
    - └─ Production [MANDATORY if Build Environments]
  - └─ Build Types [MANDATORY if Build Configuration]
    - └─ Debug [MANDATORY if Build Types]
    - └─ Release [MANDATORY if Build Types]
- └─ Build Variants [OPTIONAL]

└─ 13.3 Continuous Integration [MANDATORY]

- └─ CI Platform [MANDATORY if CI]
- └─ Automated Builds [MANDATORY if CI]
  - └─ On Commit [OR]
  - └─ On Pull Request [OR]
  - └─ Scheduled Builds [OR]
- └─ Code Quality Gates [MANDATORY if CI]
  - └─ Linting [MANDATORY if Quality Gates]
  - └─ Code Formatting [OPTIONAL]
  - └─ Static Analysis [MANDATORY if Quality Gates]
  - └─ Security Scanning [OPTIONAL]
- └─ Automated Testing [OPTIONAL]
  - └─ Unit Tests [OR]
  - └─ Integration Tests [OR]
  - └─ UI Tests [OR]
- └─ Build Artifacts [MANDATORY if CI]
  - └─ Artifact Storage [MANDATORY if Build Artifacts]
  - └─ Artifact Versioning [MANDATORY if Build Artifacts]

└─ 13.4 Continuous Deployment [OPTIONAL]

- └─ App Distribution [OPTIONAL]
  - └─ Internal Distribution [OR]
  - └─ Beta Distribution [OR]

- └─ Production Distribution [OR]
- └─ Staged Rollout [OPTIONAL]
- └─ Percentage Rollout [OR]
- └─ Canary Deployment [OR]
- └─ Rollback Strategy [OPTIONAL]

#### └─ 13.5 Environment Management [MANDATORY]

- └─ Environment Configuration [MANDATORY if Environment Management]
- └─ Environment Variables [MANDATORY if Environment Config]
- └─ Feature Flags [OPTIONAL]
- └─ Build Configurations [MANDATORY if Environment Config]
- └─ Secrets Management [MANDATORY if Environment Management]
- └─ API Keys [MANDATORY if Secrets Management]
- └─ Certificates [MANDATORY if Secrets Management]
- └─ Signing Keys [MANDATORY if Secrets Management]

#### └─ 13.6 Infrastructure as Code [OPTIONAL]

- └─ Build Pipeline as Code [OR]
- └─ Infrastructure Scripts [OR]
- └─ Configuration as Code [OR]

### └─ 14. DEPLOYMENT & DISTRIBUTION [MANDATORY]

#### └─ 14.1 App Store Deployment [OPTIONAL]

- └─ iOS App Store [OPTIONAL]
- └─ App Store Connect [MANDATORY if iOS App Store]
- └─ TestFlight [OPTIONAL]
- └─ App Store Optimization (ASO) [OPTIONAL]
- └─ Google Play Store [OPTIONAL]
- └─ Play Console [MANDATORY if Google Play]
- └─ Internal/Closed/Open Testing [OPTIONAL]
- └─ Play Store Optimization [OPTIONAL]

#### └─ 14.2 Release Management [MANDATORY]

- └─ Versioning Strategy [MANDATORY if Release Management]
- └─ Semantic Versioning [MANDATORY if Versioning]
- └─ Build Number [MANDATORY if Versioning]
- └─ Version Code/String [MANDATORY if Versioning]
- └─ Release Documentation [MANDATORY if Release Management]
- └─ Release Notes [MANDATORY if Release Documentation]
- └─ Changelog [OPTIONAL]
- └─ Migration Guide [OPTIONAL]
- └─ Release Process [MANDATORY if Release Management]
- └─ Release Approval [OPTIONAL]
- └─ Release Testing [OPTIONAL]
- └─ Release Communication [OPTIONAL]

- └─ Hotfix Process [OPTIONAL]

- └─ 14.3 App Signing [MANDATORY]

- └─ Signing Strategy [MANDATORY if App Signing - ALTERNATIVE]

- └─ Manual Signing [ALTERNATIVE]

- └─ Automated Signing [ALTERNATIVE]

- └─ Certificate Management [MANDATORY if App Signing]

- └─ Development Certificates [MANDATORY if Certificate Management]

- └─ Distribution Certificates [MANDATORY if Certificate Management]

- └─ Certificate Rotation [OPTIONAL]

- └─ 14.4 Over-The-Air (OTA) Updates [OPTIONAL]

- └─ Code Push [OR]

- └─ Asset Updates [OR]

- └─ Partial Updates [OR]

- └─ 14.5 Alternative Distribution [OPTIONAL]

- └─ Enterprise Distribution [OPTIONAL]

- └─ Apple Business Manager [OR]

- └─ Google Play Managed [OR]

- └─ MDM Integration [OR]

- └─ Third-Party Stores [OPTIONAL]

- └─ Amazon Appstore [OR]

- └─ Samsung Galaxy Store [OR]

- └─ Huawei AppGallery [OR]

- └─ Direct Download [OPTIONAL]

- └─ APK Distribution [OR]

- └─ IPA Distribution [OR]

- └─ 14.6 Post-Deployment [OPTIONAL]

- └─ Deployment Monitoring [OR]

- └─ Rollout Tracking [OR]

- └─ User Adoption Metrics [OR]

- └─ 15. FEATURE MANAGEMENT & EXPERIMENTATION [OPTIONAL]

- └─ 15.1 Feature Flags [OPTIONAL]

- └─ Feature Toggle System [MANDATORY if Feature Flags]

- └─ Remote Configuration [OPTIONAL]

- └─ Feature Targeting [OPTIONAL]

- └─ User-based Targeting [OR]

- └─ Percentage Targeting [OR]

- └─ Segment Targeting [OR]

- └─ Kill Switch [OPTIONAL]

- └─ 15.2 A/B Testing [OPTIONAL]

- └─ Experiment Design [OPTIONAL]
  - └─ Variant Configuration [MANDATORY if Experiment Design]
  - └─ Traffic Allocation [MANDATORY if Experiment Design]
  - └─ Control Group [MANDATORY if Experiment Design]
- └─ Metrics Collection [OPTIONAL]
- └─ Statistical Analysis [OPTIONAL]
- └─ Experiment Lifecycle [OPTIONAL]

### 15.3 Dynamic Content [OPTIONAL]

- └─ Remote Content Management [OR]
- └─ Server-Driven UI [OR]
- └─ Content Personalization [OR]

## 16. LOCALIZATION & INTERNATIONALIZATION (i18n/l10n) [OPTIONAL]

### 16.1 Language Support [OPTIONAL]

- └─ String Resources [MANDATORY if Language Support]
- └─ Multi-Language Support [OPTIONAL]
- └─ Language Detection [OPTIONAL]
- └─ Language Switching [OPTIONAL]
- └─ Pluralization Rules [OPTIONAL]

### 16.2 Regional Formatting [OPTIONAL]

- └─ Date & Time Formatting [OPTIONAL]
  - └─ Date Format [OR]
  - └─ Time Format [OR]
  - └─ Time Zone Support [OR]
- └─ Number Formatting [OPTIONAL]
  - └─ Decimal Separators [OR]
  - └─ Thousand Separators [OR]
- └─ Currency Formatting [OPTIONAL]
  - └─ Currency Symbols [OR]
  - └─ Currency Conversion [OR]

### 16.3 Content Localization [OPTIONAL]

- └─ Text Direction [OPTIONAL - ALTERNATIVE]
  - └─ LTR (Left-to-Right) [ALTERNATIVE]
  - └─ RTL (Right-to-Left) [ALTERNATIVE]
- └─ Localized Assets [OPTIONAL]
  - └─ Images [OR]
  - └─ Media [OR]
- └─ Cultural Adaptation [OPTIONAL]

### 16.4 Translation Management [OPTIONAL]

- └─ Translation Workflow [OR]
- └─ Translation Memory [OR]

- └─ Machine Translation [OR]

- └─ 17. DEPENDENCY MANAGEMENT [MANDATORY]

- └─ 17.1 Package Management [MANDATORY]

- └─ Package Manager [MANDATORY if Package Management]
- └─ Dependency Resolution [MANDATORY if Package Management]
- └─ Version Pinning [OPTIONAL]
- └─ Lock Files [MANDATORY if Package Management]

- └─ 17.2 Dependency Injection [OPTIONAL]

- └─ DI Framework [MANDATORY if Dependency Injection]
- └─ Dependency Graph [OPTIONAL]
- └─ Scoped Dependencies [OPTIONAL]

- └─ 17.3 Dependency Security & Compliance [MANDATORY]

- └─ Vulnerability Scanning [OPTIONAL]
- └─ License Compliance [OPTIONAL]
- └─ Dependency Updates [MANDATORY if Dependency Security]
  - └─ Automated Updates [ALTERNATIVE]
  - └─ Manual Updates [ALTERNATIVE]
- └─ Supply Chain Security [OPTIONAL]

- └─ 18. ERROR HANDLING & RESILIENCE [MANDATORY]

- └─ 18.1 Error Handling Strategy [MANDATORY]

- └─ Exception Handling [MANDATORY if Error Handling]
  - └─ Try-Catch Blocks [MANDATORY if Exception Handling]
  - └─ Error Boundaries [OPTIONAL]
  - └─ Global Error Handler [MANDATORY if Exception Handling]
- └─ Error Propagation [MANDATORY if Error Handling]
  - └─ Error Wrapping [OR]
  - └─ Error Transformation [OR]

- └─ 18.2 User Error Communication [MANDATORY]

- └─ Error Messages [MANDATORY if User Communication]
  - └─ User-Friendly Messages [MANDATORY if Error Messages]
  - └─ Localized Messages [OPTIONAL]
- └─ Error UI [OPTIONAL]
  - └─ Error Dialogs [OR]
  - └─ Error Screens [OR]
  - └─ Inline Errors [OR]
- └─ Retry Mechanisms [OPTIONAL]
- └─ Fallback UI [OPTIONAL]

- └─ 18.3 Network Error Handling [MANDATORY if Networking]

- └─ Timeout Handling [MANDATORY if Network Error Handling]

- └─ Offline Detection [MANDATORY if Network Error Handling]
- └─ Network Error Recovery [MANDATORY if Network Error Handling]
- └─ Request Retry [OPTIONAL]
  - └─ Exponential Backoff [ALTERNATIVE]
  - └─ Linear Backoff [ALTERNATIVE]
  - └─ Custom Retry Logic [ALTERNATIVE]

#### └─ 18.4 Resilience Patterns [OPTIONAL]

- └─ Circuit Breaker [OR]
- └─ Bulkhead [OR]
- └─ Timeout [OR]
- └─ Fallback [OR]

#### └─ 18.5 Data Validation [MANDATORY]

- └─ Input Validation [MANDATORY if Data Validation]
  - └─ Client-Side Validation [MANDATORY if Input Validation]
  - └─ Format Validation [MANDATORY if Input Validation]
- └─ Business Rule Validation [MANDATORY if Data Validation]
- └─ Schema Validation [OPTIONAL]

#### └─ 19. DOMAIN-SPECIFIC EXTENSIONS [OPTIONAL]

- └─ Domain Extension Reference [OPTIONAL]
  - └─ Financial Services Domain [REFERENCE to DSEM]
  - └─ E-Commerce & Retail Domain [REFERENCE to DSEM]
  - └─ Healthcare & Telemedicine Domain [REFERENCE to DSEM]
  - └─ Education & E-Learning Domain [REFERENCE to DSEM]
  - └─ Transportation & Logistics Domain [REFERENCE to DSEM]
  - └─ IoT & Smart Home Domain [REFERENCE to DSEM]

## 10.5. DOMAIN-SPECIFIC EXTENSIONS MODEL (DSEM)

### Domain-Specific Mobile Features [ROOT - OPTIONAL]

[REQUIRES: Mobile Application Architecture (MAFM)]

[CONSTRAINT: Select at most ONE domain per product configuration]

#### 1. FINANCIAL SERVICES DOMAIN [OPTIONAL - ALTERNATIVE]

##### 1.1 Account Management [OPTIONAL]

###### Account Overview [OPTIONAL]

Balance Display [MANDATORY if Account Overview]

Account Details [MANDATORY if Account Overview]

Multi-Account Support [OPTIONAL]

###### Account Operations [OPTIONAL]

Account Opening [OPTIONAL]

Account Closure [OPTIONAL]

Account Settings [OPTIONAL]

###### Statement Management [OPTIONAL]

Statement Download [OPTIONAL]

Statement History [OPTIONAL]

eStatement Enrollment [OPTIONAL]

##### 1.2 Transactions [OPTIONAL]

###### Transaction History [OPTIONAL]

Transaction List [MANDATORY if Transaction History]

Transaction Details [MANDATORY if Transaction History]

Transaction Search [OPTIONAL]

Transaction Filtering [OPTIONAL]

###### Money Transfer [OPTIONAL]

Internal Transfer [OPTIONAL]

External Transfer [OPTIONAL]

ACH Transfer [OPTIONAL]

Wire Transfer [OPTIONAL]

Instant Transfer [OPTIONAL]

International Transfer [OPTIONAL]

SWIFT [OPTIONAL]

SEPA [OPTIONAL]

P2P Transfer [OPTIONAL]

###### Bill Payment [OPTIONAL]

Biller Management [OPTIONAL]

One-Time Payment [OPTIONAL]

Recurring Payment [OPTIONAL]

Payment Scheduling [OPTIONAL]

Mobile Check Deposit [OPTIONAL]

- └─ Check Capture [MANDATORY if Mobile Check]
- └─ Check Validation [OPTIONAL]
- └─ Deposit Limits [OPTIONAL]

### └─ 1.3 Cards & Payment Methods [OPTIONAL]

- └─ Card Management [OPTIONAL]
  - └─ Card List [OPTIONAL]
  - └─ Card Details [OPTIONAL]
  - └─ Card Activation [OPTIONAL]
  - └─ Card Lock/Unlock [OPTIONAL]
  - └─ Card PIN Management [OPTIONAL]
  - └─ Card Replacement [OPTIONAL]
- └─ Virtual Cards [OPTIONAL]
  - └─ Virtual Card Creation [OPTIONAL]
  - └─ Single-Use Cards [OPTIONAL]
- └─ Contactless Payment [OPTIONAL]
  - └─ NFC Payment [OPTIONAL]
  - └─ QR Code Payment [OPTIONAL]

### └─ 1.4 Lending & Credit [OPTIONAL]

- └─ Loan Management [OPTIONAL]
  - └─ Loan Overview [OPTIONAL]
  - └─ Loan Details [OPTIONAL]
  - └─ Loan Payment [OPTIONAL]
  - └─ Early Payoff Calculator [OPTIONAL]
- └─ Loan Application [OPTIONAL]
  - └─ Loan Eligibility Check [OPTIONAL]
  - └─ Loan Calculator [OPTIONAL]
  - └─ Application Form [OPTIONAL]
  - └─ Document Upload [OPTIONAL]
  - └─ Application Tracking [OPTIONAL]
- └─ Credit Card Services [OPTIONAL]
  - └─ Credit Card Application [OPTIONAL]
  - └─ Credit Limit Management [OPTIONAL]
  - └─ Credit Card Rewards [OPTIONAL]
  - └─ Statement Generation [OPTIONAL]
- └─ Credit Score [OPTIONAL]
  - └─ Credit Score Display [OPTIONAL]
  - └─ Credit Monitoring [OPTIONAL]
  - └─ Credit Score Factors [OPTIONAL]

### └─ 1.5 Investment & Wealth Management [OPTIONAL]

- └─ Portfolio Management [OPTIONAL]
  - └─ Portfolio Overview [OPTIONAL]
  - └─ Asset Allocation [OPTIONAL]

- └─ Performance Tracking [OPTIONAL]
- └─ Portfolio Analysis [OPTIONAL]
- └─ Trading [OPTIONAL]
  - └─ Stock Trading [OPTIONAL]
  - └─ Mutual Funds [OPTIONAL]
  - └─ ETFs [OPTIONAL]
  - └─ Bonds [OPTIONAL]
  - └─ Order Management [OPTIONAL]
- └─ Market Data [OPTIONAL]
  - └─ Real-time Quotes [OPTIONAL]
  - └─ Market News [OPTIONAL]
  - └─ Charts & Analytics [OPTIONAL]
  - └─ Watchlist [OPTIONAL]
- └─ Robo-Advisory [OPTIONAL]
  - └─ Risk Assessment [OPTIONAL]
  - └─ Automated Rebalancing [OPTIONAL]
  - └─ Investment Recommendations [OPTIONAL]

#### └─ 1.6 Insurance Services [OPTIONAL]

- └─ Policy Management [OPTIONAL]
  - └─ Policy Overview [OPTIONAL]
  - └─ Policy Details [OPTIONAL]
  - └─ Policy Documents [OPTIONAL]
- └─ Claims Management [OPTIONAL]
  - └─ Claim Filing [OPTIONAL]
  - └─ Claim Tracking [OPTIONAL]
  - └─ Claim Documentation [OPTIONAL]
- └─ Premium Payment [OPTIONAL]

#### └─ 1.7 Budgeting & Financial Planning [OPTIONAL]

- └─ Budget Management [OPTIONAL]
  - └─ Budget Creation [OPTIONAL]
  - └─ Budget Tracking [OPTIONAL]
  - └─ Budget Alerts [OPTIONAL]
- └─ Expense Categorization [OPTIONAL]
  - └─ Auto-Categorization [OPTIONAL]
  - └─ Manual Categorization [OPTIONAL]
- └─ Spending Insights [OPTIONAL]
  - └─ Spending Reports [OPTIONAL]
  - └─ Spending Trends [OPTIONAL]
  - └─ Merchant Analysis [OPTIONAL]
- └─ Financial Goals [OPTIONAL]
  - └─ Goal Setting [OPTIONAL]
  - └─ Goal Tracking [OPTIONAL]
  - └─ Savings Recommendations [OPTIONAL]

└─ 1.8 Alerts & Notifications [OPTIONAL]

- └─ Transaction Alerts [OPTIONAL]
- └─ Balance Alerts [OPTIONAL]
- └─ Payment Due Alerts [OPTIONAL]
- └─ Security Alerts [OPTIONAL]
- └─ Promotional Alerts [OPTIONAL]

└─ 1.9 Regulatory Compliance [MANDATORY if Financial Services]

- └─ KYC (Know Your Customer) [MANDATORY]
  - └─ Identity Verification [MANDATORY]
  - └─ Document Verification [MANDATORY]
  - └─ Biometric Verification [OPTIONAL]
- └─ AML (Anti-Money Laundering) [MANDATORY]
  - └─ Transaction Monitoring [MANDATORY]
  - └─ Suspicious Activity Reporting [MANDATORY]
  - └─ Risk Scoring [OPTIONAL]
- └─ PCI DSS Compliance [MANDATORY if Card Processing]
- └─ SOX Compliance [OPTIONAL]
- └─ GLBA Compliance [OPTIONAL]
- └─ Financial Reporting [OPTIONAL]
  - └─ Tax Documents [OPTIONAL]
  - └─ 1099 Forms [OPTIONAL]
  - └─ Annual Reports [OPTIONAL]

└─ 2. E-COMMERCE & RETAIL DOMAIN [OPTIONAL - ALTERNATIVE]

└─ 2.1 Product Catalog [OPTIONAL]

- └─ Product Discovery [OPTIONAL]
  - └─ Product List Page [MANDATORY if Product Catalog]
    - └─ Grid View [ALTERNATIVE]
    - └─ List View [ALTERNATIVE]
  - └─ Product Detail Page [MANDATORY if Product Catalog]
    - └─ Product Images [MANDATORY]
    - └─ Product Gallery [OPTIONAL]
    - └─ 360° View [OPTIONAL]
    - └─ AR Preview [OPTIONAL]
    - └─ Product Description [MANDATORY]
    - └─ Product Specifications [OPTIONAL]
    - └─ Variants Selection [OPTIONAL]
    - └─ Availability Status [MANDATORY]
  - └─ Product Categories [OPTIONAL]
    - └─ Category Navigation [OPTIONAL]
    - └─ Subcategories [OPTIONAL]
    - └─ Category Filters [OPTIONAL]
- └─ Product Search [OPTIONAL]

- └─ Text Search [MANDATORY if Product Search]
- └─ Voice Search [OPTIONAL]
- └─ Visual Search [OPTIONAL]
  - └─ Image Upload Search [OPTIONAL]
    - └─ Camera Search [OPTIONAL]
- └─ Barcode/QR Scanner [OPTIONAL]
- └─ Search Autocomplete [OPTIONAL]
- └─ Search Suggestions [OPTIONAL]
- └─ Search History [OPTIONAL]
- └─ Filtering & Sorting [OPTIONAL]
  - └─ Price Range Filter [OPTIONAL]
  - └─ Brand Filter [OPTIONAL]
  - └─ Size/Color Filter [OPTIONAL]
  - └─ Rating Filter [OPTIONAL]
  - └─ Availability Filter [OPTIONAL]
  - └─ Sort Options [OPTIONAL]
    - └─ Price (Low-High) [OPTIONAL]
    - └─ Price (High-Low) [OPTIONAL]
    - └─ Newest [OPTIONAL]
    - └─ Bestseller [OPTIONAL]
    - └─ Rating [OPTIONAL]
- └─ Product Recommendations [OPTIONAL]
  - └─ Personalized Recommendations [OPTIONAL]
  - └─ Similar Products [OPTIONAL]
  - └─ Frequently Bought Together [OPTIONAL]
  - └─ Recently Viewed [OPTIONAL]

## └─ 2.2 Shopping Cart [OPTIONAL]

- └─ Cart Management [MANDATORY if Shopping Cart]
  - └─ Add to Cart [MANDATORY]
  - └─ Update Quantity [MANDATORY]
  - └─ Remove from Cart [MANDATORY]
  - └─ Save for Later [OPTIONAL]
  - └─ Move to Wishlist [OPTIONAL]
- └─ Cart Display [MANDATORY if Shopping Cart]
  - └─ Item Details [MANDATORY]
  - └─ Price Breakdown [MANDATORY]
  - └─ Subtotal [MANDATORY]
  - └─ Taxes [OPTIONAL]
  - └─ Shipping Cost [OPTIONAL]
  - └─ Total [MANDATORY]
- └─ Cart Persistence [OPTIONAL]
  - └─ Guest Cart [OPTIONAL]
  - └─ Logged-in Cart Sync [OPTIONAL]

- └─ Mini Cart [OPTIONAL]

- └─ 2.3 Checkout Process [OPTIONAL]

- └─ Guest Checkout [OPTIONAL]

- └─ Express Checkout [OPTIONAL]

- └─ One-Click Purchase [OPTIONAL]

- └─ Buy Now [OPTIONAL]

- └─ Shipping Information [MANDATORY if Physical Products]

- └─ Address Management [MANDATORY]

- └─ Add New Address [OPTIONAL]

- └─ Edit Address [OPTIONAL]

- └─ Delete Address [OPTIONAL]

- └─ Default Address [OPTIONAL]

- └─ Shipping Methods [MANDATORY]

- └─ Standard Shipping [OPTIONAL]

- └─ Express Shipping [OPTIONAL]

- └─ Same-Day Delivery [OPTIONAL]

- └─ Store Pickup [OPTIONAL]

- └─ Delivery Time Estimation [OPTIONAL]

- └─ Payment Methods [MANDATORY if Checkout]

- └─ Credit/Debit Card [OPTIONAL]

- └─ Digital Wallets [OPTIONAL]

- └─ Apple Pay [OPTIONAL]

- └─ Google Pay [OPTIONAL]

- └─ PayPal [OPTIONAL]

- └─ Shop Pay [OPTIONAL]

- └─ Buy Now Pay Later [OPTIONAL]

- └─ Klarna [OPTIONAL]

- └─ Affirm [OPTIONAL]

- └─ Afterpay [OPTIONAL]

- └─ Bank Transfer [OPTIONAL]

- └─ Cash on Delivery [OPTIONAL]

- └─ Gift Cards [OPTIONAL]

- └─ Order Review [MANDATORY if Checkout]

- └─ Order Summary [MANDATORY]

- └─ Terms & Conditions [MANDATORY]

- └─ Privacy Policy Acceptance [OPTIONAL]

- └─ Order Confirmation [MANDATORY if Checkout]

- └─ Order Number [MANDATORY]

- └─ Order Receipt [MANDATORY]

- └─ Email Confirmation [OPTIONAL]

- └─ 2.4 Order Management [OPTIONAL]

- └─ Order History [OPTIONAL]

- └─ Order List [MANDATORY if Order Management]

- └─ Order Details [MANDATORY if Order Management]
- └─ Order Status [MANDATORY]
- └─ Order Tracking [OPTIONAL]
  - └─ Real-time Tracking [OPTIONAL]
  - └─ Tracking Number [OPTIONAL]
  - └─ Delivery Updates [OPTIONAL]
- └─ Order Actions [OPTIONAL]
  - └─ Cancel Order [OPTIONAL]
  - └─ Modify Order [OPTIONAL]
  - └─ Reorder [OPTIONAL]
- └─ Returns & Refunds [OPTIONAL]
  - └─ Return Request [OPTIONAL]
  - └─ Return Label [OPTIONAL]
  - └─ Return Tracking [OPTIONAL]
  - └─ Refund Processing [OPTIONAL]
  - └─ Exchange [OPTIONAL]
- └─ Invoice & Receipt [OPTIONAL]
  - └─ Invoice Download [OPTIONAL]
  - └─ Receipt Email [OPTIONAL]

## └─ 2.5 Inventory Management [OPTIONAL]

- └─ Stock Availability [OPTIONAL]
  - └─ In-Stock Status [MANDATORY if Inventory]
  - └─ Out-of-Stock Alerts [OPTIONAL]
  - └─ Low-Stock Indicators [OPTIONAL]
  - └─ Back-in-Stock Notifications [OPTIONAL]
- └─ Store Inventory [OPTIONAL]
  - └─ Store Locator [OPTIONAL]
  - └─ Store Availability Check [OPTIONAL]
- └─ Inventory Sync [OPTIONAL]

## └─ 2.6 Wishlist & Favorites [OPTIONAL]

- └─ Wishlist Management [OPTIONAL]
  - └─ Add to Wishlist [MANDATORY if Wishlist]
  - └─ Remove from Wishlist [MANDATORY if Wishlist]
  - └─ Move to Cart [OPTIONAL]
  - └─ Share Wishlist [OPTIONAL]
- └─ Multiple Wishlists [OPTIONAL]
- └─ Wishlist Notifications [OPTIONAL]
  - └─ Price Drop Alerts [OPTIONAL]
  - └─ Back-in-Stock Alerts [OPTIONAL]

## └─ 2.7 Reviews & Ratings [OPTIONAL]

- └─ Product Reviews [OPTIONAL]
  - └─ Review Display [MANDATORY if Reviews]

- Star Rating [MANDATORY]
    - Review Text [OPTIONAL]
    - Review Images [OPTIONAL]
    - Verified Purchase Badge [OPTIONAL]
  - Write Review [OPTIONAL]
    - Rating Selection [MANDATORY]
    - Review Title [OPTIONAL]
    - Review Text [MANDATORY]
    - Photo Upload [OPTIONAL]
  - Review Filtering [OPTIONAL]
    - By Rating [OPTIONAL]
    - By Date [OPTIONAL]
    - By Verified Purchase [OPTIONAL]
  - Review Voting [OPTIONAL]
    - Helpful Vote [OPTIONAL]
    - Report Review [OPTIONAL]
  - Q&A Section [OPTIONAL]
    - Ask Question [OPTIONAL]
    - Answer Question [OPTIONAL]

## — 2.8 Promotions & Discounts [OPTIONAL]

- Coupon Management [OPTIONAL]
  - Coupon Code Entry [OPTIONAL]
  - Auto-Apply Coupons [OPTIONAL]
  - Coupon Wallet [OPTIONAL]
- Promotional Campaigns [OPTIONAL]
  - Flash Sales [OPTIONAL]
  - Daily Deals [OPTIONAL]
  - Bundle Offers [OPTIONAL]
  - Clearance Section [OPTIONAL]
- Discount Display [OPTIONAL]
  - Original Price [OPTIONAL]
  - Discounted Price [OPTIONAL]
  - Savings Amount [OPTIONAL]
- Time-Limited Offers [OPTIONAL]
  - Countdown Timer [OPTIONAL]

## — 2.9 Loyalty Programs [OPTIONAL]

- Points System [OPTIONAL]
  - Earn Points [OPTIONAL]
  - Redeem Points [OPTIONAL]
  - Points Balance [OPTIONAL]
  - Points History [OPTIONAL]
- Membership Tiers [OPTIONAL]
  - Tier Benefits [OPTIONAL]

- └─ Tier Progress [OPTIONAL]
- └─ Tier Rewards [OPTIONAL]
- └─ Referral Program [OPTIONAL]
  - └─ Referral Code [OPTIONAL]
  - └─ Referral Tracking [OPTIONAL]
  - └─ Referral Rewards [OPTIONAL]
- └─ Exclusive Offers [OPTIONAL]

#### └─ 2.10 Customer Support [OPTIONAL]

- └─ Help Center [OPTIONAL]
  - └─ FAQs [OPTIONAL]
  - └─ Knowledge Base [OPTIONAL]
- └─ Live Chat [OPTIONAL]
  - └─ Text Chat [OPTIONAL]
  - └─ Chatbot [OPTIONAL]
- └─ Contact Options [OPTIONAL]
  - └─ Email Support [OPTIONAL]
  - └─ Phone Support [OPTIONAL]
  - └─ Contact Form [OPTIONAL]
- └─ Order Support [OPTIONAL]
  - └─ Order Inquiry [OPTIONAL]
  - └─ Complaint Management [OPTIONAL]

#### └─ 2.11 Social Shopping [OPTIONAL]

- └─ Social Sharing [OPTIONAL]
  - └─ Share Products [OPTIONAL]
  - └─ Share Wishlist [OPTIONAL]
- └─ Social Proof [OPTIONAL]
  - └─ Recently Purchased [OPTIONAL]
  - └─ Popular Products [OPTIONAL]
  - └─ Trending Items [OPTIONAL]
- └─ User-Generated Content [OPTIONAL]
  - └─ Customer Photos [OPTIONAL]
  - └─ Customer Videos [OPTIONAL]

### └─ 3. HEALTHCARE & TELEMEDICINE DOMAIN [OPTIONAL - ALTERNATIVE]

#### └─ 3.1 Patient Profile [OPTIONAL]

- └─ Personal Information [MANDATORY if Healthcare]
  - └─ Demographics [MANDATORY]
  - └─ Contact Information [MANDATORY]
  - └─ Emergency Contacts [OPTIONAL]
- └─ Medical History [OPTIONAL]
  - └─ Past Conditions [OPTIONAL]
  - └─ Surgeries [OPTIONAL]
  - └─ Chronic Conditions [OPTIONAL]

- └─ Family History [OPTIONAL]
- └─ Allergies [OPTIONAL]
  - └─ Drug Allergies [OPTIONAL]
  - └─ Food Allergies [OPTIONAL]
  - └─ Environmental Allergies [OPTIONAL]
- └─ Medications [OPTIONAL]
  - └─ Current Medications [OPTIONAL]
  - └─ Past Medications [OPTIONAL]
  - └─ Medication Schedule [OPTIONAL]
- └─ Insurance Information [OPTIONAL]
  - └─ Primary Insurance [OPTIONAL]
  - └─ Secondary Insurance [OPTIONAL]
  - └─ Insurance Card Upload [OPTIONAL]

### └─ 3.2 Appointments [OPTIONAL]

- └─ Appointment Scheduling [OPTIONAL]
  - └─ Doctor Search [OPTIONAL]
    - └─ By Specialty [OPTIONAL]
    - └─ By Location [OPTIONAL]
    - └─ By Availability [OPTIONAL]
    - └─ By Rating [OPTIONAL]
  - └─ Calendar View [OPTIONAL]
  - └─ Slot Selection [MANDATORY if Scheduling]
  - └─ Reason for Visit [OPTIONAL]
  - └─ Appointment Confirmation [MANDATORY]
- └─ Appointment Management [OPTIONAL]
  - └─ View Appointments [MANDATORY if Appointments]
  - └─ Reschedule [OPTIONAL]
  - └─ Cancel Appointment [OPTIONAL]
  - └─ Appointment History [OPTIONAL]
- └─ Appointment Reminders [OPTIONAL]
  - └─ Push Notifications [OPTIONAL]
  - └─ SMS Reminders [OPTIONAL]
  - └─ Email Reminders [OPTIONAL]
- └─ Check-in [OPTIONAL]
  - └─ Pre-Appointment Forms [OPTIONAL]
  - └─ Digital Check-in [OPTIONAL]
  - └─ Waiting Room Status [OPTIONAL]

### └─ 3.3 Telemedicine [OPTIONAL]

- └─ Virtual Consultation [OPTIONAL]
  - └─ Video Call [MANDATORY if Telemedicine]
  - └─ Audio Call [OPTIONAL]
  - └─ Text Chat [OPTIONAL]
  - └─ Screen Sharing [OPTIONAL]

- └─ Consultation Preparation [OPTIONAL]
  - └─ Symptom Checker [OPTIONAL]
  - └─ Pre-Consultation Form [OPTIONAL]
  - └─ Document Upload [OPTIONAL]
- └─ Consultation Features [OPTIONAL]
  - └─ Waiting Room [OPTIONAL]
  - └─ Session Recording [OPTIONAL]
  - └─ Prescription Writing [OPTIONAL]
  - └─ Lab Order Creation [OPTIONAL]
- └─ Post-Consultation [OPTIONAL]
  - └─ Visit Summary [OPTIONAL]
  - └─ Prescription Download [OPTIONAL]
  - └─ Follow-up Scheduling [OPTIONAL]

### └─ 3.4 Electronic Health Records (EHR) [OPTIONAL]

- └─ Medical Records [OPTIONAL]
  - └─ Lab Results [OPTIONAL]
  - └─ Imaging Reports [OPTIONAL]
  - └─ Visit Notes [OPTIONAL]
  - └─ Discharge Summaries [OPTIONAL]
  - └─ Immunization Records [OPTIONAL]
- └─ Record Sharing [OPTIONAL]
  - └─ Share with Provider [OPTIONAL]
  - └─ Share with Family [OPTIONAL]
  - └─ Export Records [OPTIONAL]
- └─ Document Management [OPTIONAL]
  - └─ Upload Documents [OPTIONAL]
  - └─ Organize Documents [OPTIONAL]
  - └─ Document Viewer [OPTIONAL]

### └─ 3.5 Prescription Management [OPTIONAL]

- └─ ePrescription [OPTIONAL]
  - └─ Prescription List [MANDATORY if Prescriptions]
  - └─ Prescription Details [OPTIONAL]
  - └─ Refill Request [OPTIONAL]
  - └─ Prescription Renewal [OPTIONAL]
- └─ Pharmacy Integration [OPTIONAL]
  - └─ Pharmacy Locator [OPTIONAL]
  - └─ Prescription Transfer [OPTIONAL]
  - └─ Prescription Delivery [OPTIONAL]
- └─ Medication Reminders [OPTIONAL]
  - └─ Dosage Reminders [OPTIONAL]
  - └─ Refill Reminders [OPTIONAL]
  - └─ Medication Tracking [OPTIONAL]

- └─ 3.6 Health Tracking [OPTIONAL]
  - └─ Vital Signs [OPTIONAL]
    - └─ Blood Pressure [OPTIONAL]
    - └─ Heart Rate [OPTIONAL]
    - └─ Blood Glucose [OPTIONAL]
    - └─ Temperature [OPTIONAL]
    - └─ Weight [OPTIONAL]
    - └─ Oxygen Saturation [OPTIONAL]
  - └─ Activity Tracking [OPTIONAL]
    - └─ Steps [OPTIONAL]
    - └─ Exercise [OPTIONAL]
    - └─ Sleep [OPTIONAL]
    - └─ Calories [OPTIONAL]
  - └─ Symptoms Tracking [OPTIONAL]
    - └─ Symptom Log [OPTIONAL]
    - └─ Pain Level [OPTIONAL]
    - └─ Mood Tracking [OPTIONAL]
  - └─ Health Goals [OPTIONAL]
    - └─ Goal Setting [OPTIONAL]
    - └─ Progress Tracking [OPTIONAL]
    - └─ Goal Reminders [OPTIONAL]
  - └─ Device Integration [OPTIONAL]
    - └─ Wearable Devices [OPTIONAL]
    - └─ Smart Scales [OPTIONAL]
    - └─ Blood Pressure Monitors [OPTIONAL]
    - └─ Glucometers [OPTIONAL]

- └─ 3.7 Lab & Imaging [OPTIONAL]
  - └─ Lab Tests [OPTIONAL]
    - └─ Lab Test Ordering [OPTIONAL]
    - └─ Lab Results [OPTIONAL]
    - └─ Lab History [OPTIONAL]
    - └─ Lab Interpretation [OPTIONAL]
  - └─ Imaging Services [OPTIONAL]
    - └─ Imaging Appointment [OPTIONAL]
    - └─ Imaging Results [OPTIONAL]
    - └─ Image Viewer [OPTIONAL]

- └─ 3.8 Billing & Insurance [OPTIONAL]
  - └─ Bill Management [OPTIONAL]
    - └─ View Bills [OPTIONAL]
    - └─ Bill Details [OPTIONAL]
    - └─ Payment History [OPTIONAL]
    - └─ Bill Download [OPTIONAL]

- └─ Payment Processing [OPTIONAL]
  - └─ Pay Bill [OPTIONAL]
  - └─ Payment Plans [OPTIONAL]
    - └─ Auto-Pay [OPTIONAL]
- └─ Insurance Claims [OPTIONAL]
  - └─ Claim Status [OPTIONAL]
  - └─ Claim History [OPTIONAL]
    - └─ Explanation of Benefits [OPTIONAL]
- └─ Cost Estimator [OPTIONAL]

- └─ 3.9 Health Education [OPTIONAL]
  - └─ Health Articles [OPTIONAL]
  - └─ Videos [OPTIONAL]
  - └─ Wellness Programs [OPTIONAL]
  - └─ Health Assessments [OPTIONAL]

- └─ 3.10 Compliance & Privacy [MANDATORY if Healthcare]

- └─ HIPAA Compliance [MANDATORY]
  - └─ Data Encryption [MANDATORY]
  - └─ Access Controls [MANDATORY]
  - └─ Audit Logging [MANDATORY]
    - └─ Breach Notification [MANDATORY]
- └─ Consent Management [MANDATORY]
  - └─ Informed Consent [MANDATORY]
  - └─ Data Sharing Consent [MANDATORY]
    - └─ Research Consent [OPTIONAL]
- └─ Patient Rights [MANDATORY]
  - └─ Right to Access [MANDATORY]
  - └─ Right to Amend [OPTIONAL]
    - └─ Right to Restrict [OPTIONAL]
- └─ Clinical Safety [OPTIONAL]
  - └─ Medical Disclaimer [MANDATORY]
  - └─ Emergency Information [MANDATORY]

- └─ 4. EDUCATION & E-LEARNING DOMAIN [OPTIONAL - ALTERNATIVE]

- └─ 4.1 Course Management [OPTIONAL]
  - └─ Course Catalog [OPTIONAL]
    - └─ Browse Courses [MANDATORY if Course Management]
    - └─ Course Categories [OPTIONAL]
    - └─ Course Search [OPTIONAL]
      - └─ Course Filtering [OPTIONAL]
  - └─ Course Details [OPTIONAL]
    - └─ Course Overview [MANDATORY]
    - └─ Syllabus [OPTIONAL]
    - └─ Instructor Profile [OPTIONAL]

- └─ Course Duration [MANDATORY]
- └─ Prerequisites [OPTIONAL]
- └─ Learning Outcomes [OPTIONAL]
- └─ Course Reviews [OPTIONAL]
- └─ Course Enrollment [OPTIONAL]
  - └─ Enroll in Course [MANDATORY]
  - └─ Unenroll [OPTIONAL]
  - └─ Waitlist [OPTIONAL]
  - └─ Enrollment Confirmation [OPTIONAL]
- └─ My Courses [OPTIONAL]
  - └─ Enrolled Courses [MANDATORY]
  - └─ Completed Courses [OPTIONAL]
  - └─ In-Progress Courses [OPTIONAL]
  - └─ Saved Courses [OPTIONAL]

#### └─ 4.2 Learning Content [OPTIONAL]

- └─ Video Lessons [OPTIONAL]
  - └─ Video Player [MANDATORY if Video Lessons]
  - └─ Playback Speed Control [OPTIONAL]
  - └─ Subtitles/Captions [OPTIONAL]
  - └─ Video Quality Selection [OPTIONAL]
  - └─ Offline Download [OPTIONAL]
  - └─ Resume Playback [OPTIONAL]
- └─ Reading Materials [OPTIONAL]
  - └─ Text Content [OPTIONAL]
  - └─ PDF Documents [OPTIONAL]
  - └─ Articles [OPTIONAL]
  - └─ eBooks [OPTIONAL]
- └─ Interactive Content [OPTIONAL]
  - └─ Simulations [OPTIONAL]
  - └─ Interactive Exercises [OPTIONAL]
  - └─ Code Labs [OPTIONAL]
- └─ Audio Lessons [OPTIONAL]
  - └─ Audio Player [OPTIONAL]
  - └─ Podcasts [OPTIONAL]
- └─ Live Sessions [OPTIONAL]
  - └─ Live Streaming [OPTIONAL]
  - └─ Live Chat [OPTIONAL]
  - └─ Q&A Session [OPTIONAL]

#### └─ 4.3 Assessments & Quizzes [OPTIONAL]

- └─ Quiz Types [OPTIONAL]
  - └─ Multiple Choice [OR]
  - └─ True/False [OR]

- └─ Fill-in-the-Blank [OR]
- └─ Essay Questions [OR]
- └─ Matching [OR]
- └─ Coding Challenges [OR]
- └─ Quiz Features [OPTIONAL]
  - └─ Timed Quizzes [OPTIONAL]
  - └─ Randomized Questions [OPTIONAL]
  - └─ Instant Feedback [OPTIONAL]
  - └─ Retake Attempts [OPTIONAL]
  - └─ Explanation of Answers [OPTIONAL]
- └─ Assignments [OPTIONAL]
  - └─ Assignment Submission [OPTIONAL]
  - └─ File Upload [OPTIONAL]
  - └─ Peer Review [OPTIONAL]
  - └─ Assignment Grading [OPTIONAL]
- └─ Exams [OPTIONAL]
  - └─ Proctored Exams [OPTIONAL]
  - └─ Exam Scheduling [OPTIONAL]
  - └─ Exam Results [OPTIONAL]

#### └─ 4.4 Progress Tracking [OPTIONAL]

- └─ Learning Progress [OPTIONAL]
  - └─ Course Progress [MANDATORY if Progress Tracking]
  - └─ Lesson Completion [OPTIONAL]
  - └─ Progress Dashboard [OPTIONAL]
  - └─ Progress Reports [OPTIONAL]
- └─ Performance Analytics [OPTIONAL]
  - └─ Quiz Scores [OPTIONAL]
  - └─ Assignment Grades [OPTIONAL]
  - └─ Overall Grade [OPTIONAL]
  - └─ Performance Trends [OPTIONAL]
- └─ Learning Path [OPTIONAL]
  - └─ Recommended Next Course [OPTIONAL]
  - └─ Learning Milestones [OPTIONAL]
  - └─ Skill Development [OPTIONAL]
- └─ Time Tracking [OPTIONAL]
  - └─ Time Spent [OPTIONAL]
  - └─ Study Streak [OPTIONAL]

#### └─ 4.5 Certificates & Credentials [OPTIONAL]

- └─ Certificate Generation [OPTIONAL]
  - └─ Course Completion Certificate [OPTIONAL]
  - └─ Certificate Download [OPTIONAL]
  - └─ Certificate Sharing [OPTIONAL]
- └─ Digital Badges [OPTIONAL]

- └─ Badge Collection [OPTIONAL]
- └─ Badge Sharing [OPTIONAL]
- └─ Credentials Verification [OPTIONAL]
- └─ Certificate Authenticity [OPTIONAL]

#### └─ 4.6 Social Learning [OPTIONAL]

- └─ Discussion Forums [OPTIONAL]
  - └─ Create Thread [OPTIONAL]
  - └─ Reply to Thread [OPTIONAL]
  - └─ Upvote/Downvote [OPTIONAL]
  - └─ Forum Moderation [OPTIONAL]
- └─ Study Groups [OPTIONAL]
  - └─ Create Group [OPTIONAL]
  - └─ Join Group [OPTIONAL]
  - └─ Group Chat [OPTIONAL]
  - └─ Group Activities [OPTIONAL]
- └─ Peer Interaction [OPTIONAL]
  - └─ Student Profiles [OPTIONAL]
  - └─ Direct Messaging [OPTIONAL]
  - └─ Collaborative Projects [OPTIONAL]
- └─ Leaderboards [OPTIONAL]
  - └─ Course Leaderboard [OPTIONAL]
  - └─ Global Leaderboard [OPTIONAL]

#### └─ 4.7 Instructor Features [OPTIONAL]

- └─ Course Creation [OPTIONAL]
- └─ Content Upload [OPTIONAL]
- └─ Student Management [OPTIONAL]
- └─ Grading [OPTIONAL]
- └─ Analytics Dashboard [OPTIONAL]

#### └─ 4.8 Gamification [OPTIONAL]

- └─ Points System [OPTIONAL]
- └─ Achievements [OPTIONAL]
- └─ Levels [OPTIONAL]
- └─ Rewards [OPTIONAL]
- └─ Challenges [OPTIONAL]

### └─ 5. TRANSPORTATION & LOGISTICS DOMAIN [OPTIONAL - ALTERNATIVE]

#### └─ 5.1 Ride Services [OPTIONAL]

- └─ Ride Booking [OPTIONAL]
  - └─ Location Selection [MANDATORY if Ride Booking]
    - └─ Current Location [OPTIONAL]
    - └─ Map Selection [OPTIONAL]
    - └─ Saved Locations [OPTIONAL]

- └─ Destination Entry [MANDATORY]
- └─ Ride Options [MANDATORY]
  - └─ Standard [OPTIONAL]
  - └─ Premium [OPTIONAL]
  - └─ Shared Ride [OPTIONAL]
  - └─ XL/Large Vehicle [OPTIONAL]
- └─ Price Estimate [OPTIONAL]
- └─ Scheduled Rides [OPTIONAL]
- └─ Ride Confirmation [MANDATORY]
- └─ Driver Matching [OPTIONAL]
  - └─ Finding Driver [MANDATORY if Ride Booking]
  - └─ Driver Details [OPTIONAL]
    - └─ Driver Name [MANDATORY]
    - └─ Driver Photo [OPTIONAL]
    - └─ Driver Rating [OPTIONAL]
    - └─ Vehicle Details [MANDATORY]
  - └─ Driver ETA [OPTIONAL]
- └─ Real-time Tracking [OPTIONAL]
  - └─ Driver Location [MANDATORY if Real-time Tracking]
  - └─ Route Display [OPTIONAL]
  - └─ Live ETA [OPTIONAL]
  - └─ Trip Progress [OPTIONAL]
- └─ In-Trip Features [OPTIONAL]
  - └─ Contact Driver [OPTIONAL]
  - └─ Share Trip [OPTIONAL]
  - └─ Safety Features [OPTIONAL]
    - └─ Emergency Button [OPTIONAL]
    - └─ Trusted Contacts [OPTIONAL]
  - └─ Change Destination [OPTIONAL]
- └─ Trip Completion [OPTIONAL]
  - └─ Payment Processing [MANDATORY]
  - └─ Receipt [OPTIONAL]
  - └─ Tip Driver [OPTIONAL]
  - └─ Rate Driver [OPTIONAL]

## └─ 5.2 Delivery Services [OPTIONAL]

- └─ Delivery Request [OPTIONAL]
  - └─ Pickup Location [MANDATORY if Delivery]
  - └─ Delivery Location [MANDATORY]
  - └─ Package Details [MANDATORY]
    - └─ Package Size [MANDATORY]
    - └─ Package Weight [OPTIONAL]
    - └─ Package Type [OPTIONAL]
  - └─ Delivery Options [OPTIONAL]

- └─ Standard Delivery [ALTERNATIVE]
- └─ Express Delivery [ALTERNATIVE]
- └─ Same-Day Delivery [ALTERNATIVE]
- └─ Delivery Instructions [OPTIONAL]
- └─ Courier Assignment [OPTIONAL]
- └─ Courier Details [OPTIONAL]
- └─ Pickup ETA [OPTIONAL]
- └─ Package Tracking [OPTIONAL]
- └─ Real-time Location [OPTIONAL]
- └─ Status Updates [MANDATORY if Tracking]
- └─ Delivery ETA [OPTIONAL]
- └─ Proof of Delivery [OPTIONAL]
- └─ Photo [OPTIONAL]
- └─ Signature [OPTIONAL]
- └─ Delivery History [OPTIONAL]

### └─ 5.3 Fleet Management [OPTIONAL]

- └─ Vehicle Tracking [OPTIONAL]
- └─ Fleet Map [OPTIONAL]
- └─ Vehicle Status [OPTIONAL]
- └─ Vehicle History [OPTIONAL]
- └─ Route Optimization [OPTIONAL]
- └─ Multi-Stop Routes [OPTIONAL]
- └─ Traffic-Aware Routing [OPTIONAL]
- └─ Route Efficiency [OPTIONAL]
- └─ Driver Management [OPTIONAL]
- └─ Driver Roster [OPTIONAL]
- └─ Driver Performance [OPTIONAL]
- └─ Driver Assignments [OPTIONAL]
- └─ Vehicle Maintenance [OPTIONAL]
- └─ Maintenance Schedule [OPTIONAL]
- └─ Maintenance Alerts [OPTIONAL]
- └─ Service History [OPTIONAL]

### └─ 5.4 Navigation & Routing [OPTIONAL]

- └─ Turn-by-Turn Navigation [OPTIONAL]
- └─ Voice Guidance [OPTIONAL]
- └─ Lane Guidance [OPTIONAL]
- └─ Speed Limit Display [OPTIONAL]
- └─ Traffic Information [OPTIONAL]
- └─ Real-time Traffic [OPTIONAL]
- └─ Incident Alerts [OPTIONAL]
- └─ Alternative Routes [OPTIONAL]
- └─ Offline Maps [OPTIONAL]
- └─ Points of Interest [OPTIONAL]

- └─ 5.5 Payment & Billing [OPTIONAL]
  - └─ Fare Calculation [MANDATORY if Transportation]
    - └─ Distance-based [ALTERNATIVE]
    - └─ Time-based [ALTERNATIVE]
    - └─ Fixed Rate [ALTERNATIVE]
  - └─ Payment Methods [MANDATORY]
    - └─ Cash [OPTIONAL]
    - └─ Card [OPTIONAL]
    - └─ Digital Wallet [OPTIONAL]
    - └─ Corporate Account [OPTIONAL]
  - └─ Invoicing [OPTIONAL]
    - └─ Trip Receipts [OPTIONAL]
    - └─ Monthly Invoices [OPTIONAL]
    - └─ Expense Reports [OPTIONAL]
  - └─ Promotions [OPTIONAL]
    - └─ Discount Codes [OPTIONAL]
    - └─ Referral Credits [OPTIONAL]

- └─ 5.6 Scheduling & Planning [OPTIONAL]
  - └─ Trip Planning [OPTIONAL]
    - └─ Route Planning [OPTIONAL]
    - └─ Multi-Modal Transport [OPTIONAL]
    - └─ Travel Time Estimation [OPTIONAL]
  - └─ Scheduled Trips [OPTIONAL]
    - └─ Future Bookings [OPTIONAL]
    - └─ Recurring Trips [OPTIONAL]
    - └─ Trip Reminders [OPTIONAL]
  - └─ Capacity Planning [OPTIONAL]

- └─ 6. IoT & SMART HOME DOMAIN [OPTIONAL - ALTERNATIVE]

- └─ 6.1 Device Management [OPTIONAL]
  - └─ Device Discovery [OPTIONAL]
    - └─ Auto-Discovery [OPTIONAL]
    - └─ Manual Pairing [OPTIONAL]
    - └─ QR Code Pairing [OPTIONAL]
    - └─ Bluetooth Pairing [OPTIONAL]
  - └─ Device Registration [OPTIONAL]
    - └─ Device Setup [MANDATORY if Device Management]
    - └─ Device Naming [OPTIONAL]
    - └─ Room Assignment [OPTIONAL]
    - └─ Device Grouping [OPTIONAL]
  - └─ Device List [OPTIONAL]
    - └─ All Devices View [MANDATORY if Device Management]
    - └─ Devices by Room [OPTIONAL]

- └─ Devices by Type [OPTIONAL]
    - └─ Device Search [OPTIONAL]
  - └─ Device Status [OPTIONAL]
    - └─ Online/Offline Status [MANDATORY]
    - └─ Battery Level [OPTIONAL]
    - └─ Signal Strength [OPTIONAL]
    - └─ Last Activity [OPTIONAL]
  - └─ Device Settings [OPTIONAL]
    - └─ Device Configuration [OPTIONAL]
    - └─ Firmware Updates [OPTIONAL]
    - └─ Device Removal [OPTIONAL]
    - └─ Factory Reset [OPTIONAL]
- └─ 6.2 Device Control [OPTIONAL]
  - └─ Basic Controls [OPTIONAL]
    - └─ On/Off Control [MANDATORY if Device Control]
    - └─ Dimming/Brightness [OPTIONAL]
    - └─ Color Control [OPTIONAL]
    - └─ Temperature Control [OPTIONAL]
  - └─ Advanced Controls [OPTIONAL]
    - └─ Multi-Device Control [OPTIONAL]
    - └─ Schedule Control [OPTIONAL]
    - └─ Timer Control [OPTIONAL]
  - └─ Remote Control [OPTIONAL]
    - └─ Internet Control [OPTIONAL]
    - └─ Local Network Control [OPTIONAL]
  - └─ Manual Control [OPTIONAL]
    - └─ Physical Button Control [OPTIONAL]
- └─ 6.3 Automation & Scenes [OPTIONAL]
  - └─ Scene Management [OPTIONAL]
    - └─ Create Scene [OPTIONAL]
      - └─ Scene Name [MANDATORY if Scene]
      - └─ Device Selection [MANDATORY]
      - └─ Device States [MANDATORY]
      - └─ Scene Icon [OPTIONAL]
    - └─ Edit Scene [OPTIONAL]
    - └─ Delete Scene [OPTIONAL]
    - └─ Execute Scene [MANDATORY if Scene]
    - └─ Scene Scheduling [OPTIONAL]
  - └─ Automation Rules [OPTIONAL]
    - └─ Trigger-based Automation [OPTIONAL]
      - └─ Time Trigger [OR]
      - └─ Device State Trigger [OR]
      - └─ Sensor Trigger [OR]

- └─ Location Trigger [OR]
    - └─ Weather Trigger [OR]
  - └─ Condition Logic [OPTIONAL]
    - └─ IF Conditions [OPTIONAL]
    - └─ AND/OR Logic [OPTIONAL]
    - └─ Time Constraints [OPTIONAL]
  - └─ Actions [OPTIONAL]
    - └─ Device Actions [MANDATORY if Automation]
    - └─ Scene Activation [OPTIONAL]
    - └─ Notifications [OPTIONAL]
    - └─ Delays [OPTIONAL]
  - └─ Automation Management [OPTIONAL]
    - └─ Enable/Disable [MANDATORY]
    - └─ Edit Automation [OPTIONAL]
    - └─ Delete Automation [OPTIONAL]
  - └─ Routines [OPTIONAL]
    - └─ Morning Routine [OPTIONAL]
    - └─ Night Routine [OPTIONAL]
    - └─ Away Mode [OPTIONAL]
    - └─ Custom Routines [OPTIONAL]

#### └─ 6.4 Voice Control Integration [OPTIONAL]

- └─ Voice Assistant [OPTIONAL - OR]
  - └─ Amazon Alexa [OR]
  - └─ Google Assistant [OR]
  - └─ Apple Siri [OR]
  - └─ Custom Voice Assistant [OR]
- └─ Voice Commands [OPTIONAL]
  - └─ Device Control Commands [OPTIONAL]
  - └─ Scene Activation Commands [OPTIONAL]
  - └─ Status Query Commands [OPTIONAL]
- └─ Voice Skill Management [OPTIONAL]

#### └─ 6.5 Energy Monitoring [OPTIONAL]

- └─ Energy Consumption [OPTIONAL]
  - └─ Real-time Usage [OPTIONAL]
  - └─ Historical Data [OPTIONAL]
  - └─ Device-level Tracking [OPTIONAL]
  - └─ Cost Estimation [OPTIONAL]
- └─ Energy Reports [OPTIONAL]
  - └─ Daily Reports [OPTIONAL]
  - └─ Weekly Reports [OPTIONAL]
  - └─ Monthly Reports [OPTIONAL]
  - └─ Comparative Analysis [OPTIONAL]
- └─ Energy Saving [OPTIONAL]

- └─ Eco Mode [OPTIONAL]
- └─ Usage Alerts [OPTIONAL]
- └─ Optimization Tips [OPTIONAL]

## └─ 6.6 Security Features [OPTIONAL]

- └─ Camera Integration [OPTIONAL]
  - └─ Live View [OPTIONAL]
  - └─ Recording [OPTIONAL]
    - └─ Cloud Storage [ALTERNATIVE]
    - └─ Local Storage [ALTERNATIVE]
  - └─ Motion Detection [OPTIONAL]
  - └─ Night Vision [OPTIONAL]
  - └─ Two-Way Audio [OPTIONAL]
- └─ Security Sensors [OPTIONAL]
  - └─ Door/Window Sensors [OPTIONAL]
  - └─ Motion Sensors [OPTIONAL]
  - └─ Glass Break Sensors [OPTIONAL]
  - └─ Smoke/CO Detectors [OPTIONAL]
- └─ Security System [OPTIONAL]
  - └─ Arm/Disarm [OPTIONAL]
  - └─ Security Modes [OPTIONAL]
    - └─ Home Mode [OPTIONAL]
    - └─ Away Mode [OPTIONAL]
    - └─ Night Mode [OPTIONAL]
  - └─ Alarm Events [OPTIONAL]
  - └─ Emergency Services [OPTIONAL]
- └─ Access Control [OPTIONAL]
  - └─ Smart Locks [OPTIONAL]
  - └─ Access Codes [OPTIONAL]
  - └─ Guest Access [OPTIONAL]
  - └─ Access Logs [OPTIONAL]

## └─ 6.7 Climate Control [OPTIONAL]

- └─ Thermostat Control [OPTIONAL]
  - └─ Temperature Setting [MANDATORY if Climate]
  - └─ Mode Selection [OPTIONAL]
    - └─ Heat [ALTERNATIVE]
    - └─ Cool [ALTERNATIVE]
    - └─ Auto [ALTERNATIVE]
    - └─ Fan Only [ALTERNATIVE]
  - └─ Schedule [OPTIONAL]
  - └─ Eco Mode [OPTIONAL]
- └─ Air Quality [OPTIONAL]
  - └─ Air Quality Monitor [OPTIONAL]

- └─ Air Purifier Control [OPTIONAL]
- └─ Humidifier/Dehumidifier [OPTIONAL]
- └─ Fan Control [OPTIONAL]
- └─ 6.8 Notifications & Alerts [OPTIONAL]
  - └─ Device Notifications [OPTIONAL]
    - └─ Status Change Alerts [OPTIONAL]
    - └─ Low Battery Alerts [OPTIONAL]
    - └─ Offline Alerts [OPTIONAL]
    - └─ Error Alerts [OPTIONAL]
  - └─ Security Alerts [OPTIONAL]
    - └─ Motion Detected [OPTIONAL]
    - └─ Door/Window Opened [OPTIONAL]
    - └─ Alarm Triggered [OPTIONAL]
    - └─ Camera Activity [OPTIONAL]
  - └─ Custom Alerts [OPTIONAL]
    - └─ Alert Rules [OPTIONAL]
- └─ 6.9 Integration & Compatibility [OPTIONAL]
  - └─ Smart Home Platforms [OPTIONAL - OR]
    - └─ Apple HomeKit [OR]
    - └─ Google Home [OR]
    - └─ Amazon Alexa [OR]
    - └─ Samsung SmartThings [OR]
    - └─ Custom Platform [OR]
  - └─ Device Protocols [OPTIONAL - OR]
    - └─ Zigbee [OR]
    - └─ Z-Wave [OR]
    - └─ WiFi [OR]
    - └─ Bluetooth [OR]
    - └─ Thread [OR]
    - └─ Matter [OR]
  - └─ Third-Party Integration [OPTIONAL]
    - └─ IFTTT [OPTIONAL]
    - └─ Webhooks [OPTIONAL]
    - └─ REST API [OPTIONAL]