

Elaboración de los diagramas del modelo de dominio del
proyecto GA2-220501093-AA2-EV01



Isidro J. Gallardo Navarro

Ficha:3070299

2025

Tecnología en Análisis y Desarrollo de
Software.

ADSO

Lista de chequeo a cubrir:

- identifica el tipo de diagrama apropiado para modelar el dominio.
- Diagrama con UML los artefactos del sistema diagrama de clases y de paquetes.
- Maneja herramientas de software para apoyar la elaboración de los diagramas.
- Elabora documento plantilla de casos de uso con base en estándares de documentación.
- Sigue las instrucciones para la elaboración de un diagrama de dominio del proyecto y el modelo conceptual

Resumen

El presente documento aborda la elaboración de diagramas del modelo de dominio para el proyecto "Censo Rural", una iniciativa tecnológica diseñada para optimizar la recolección y análisis de datos demográficos en zonas rurales apartadas de Colombia. La investigación se fundamenta en la metodología Extreme Programming (XP) y presenta un análisis integral de los artefactos del sistema, incluyendo diagramas de clases y de paquetes desarrollados bajo estándares UML 2.5. El estudio identifica los tipos de diagramas apropiados para el modelado del dominio, establece formatos de aceptación para la validación de artefactos, y propone una plantilla estandarizada para casos de uso. Los resultados proporcionan una base sólida para el desarrollo del software, garantizando la trazabilidad entre requisitos, diseño e implementación. Palabras clave: modelo de dominio, UML, diagramas de clases, censo rural, metodología XP, ingeniería de software

Tabla de Contenido

Introducción

Marco Metodológico

Identificación del Tipo de Diagrama Apropiado

Modelado UML de Artefactos del Sistema

Herramientas de Software Utilizadas

Plantilla de Casos de Uso

Conceptos del Dominio del Proyecto

Formatos de Aceptación

Diagrama de Dominio del Proyecto

Validación y Calidad del Modelo

Conclusiones

Referencias

Anexos

1. Introducción

1.1 Planteamiento del Problema

El desarrollo de sistemas de información para la recolección de datos censales en zonas rurales presenta desafíos únicos que requieren un modelado cuidadoso del dominio del problema. La complejidad inherente de las comunidades rurales, combinada con las limitaciones tecnológicas

y de conectividad, demanda un enfoque sistemático para la identificación y representación de los elementos fundamentales del sistema (Larman, 2004).

El proyecto "Censo Rural" surge como respuesta a la necesidad de modernizar los procesos de recolección de información demográfica en zonas apartadas de Colombia, donde los métodos tradicionales han mostrado limitaciones significativas en términos de eficiencia, precisión y cobertura (Departamento Administrativo Nacional de Estadística [DANE], 2023).

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un modelo de dominio integral para el proyecto "Censo Rural" que incluya la identificación de diagramas apropiados, el modelado UML de artefactos del sistema, y la definición de estándares de documentación que faciliten el desarrollo de una solución tecnológica robusta y escalable.

1.2.2 Objetivos Específicos

Identificar el tipo de diagrama más apropiado para modelar el dominio del proyecto "Censo Rural"

Desarrollar diagramas UML de clases y paquetes que representen la arquitectura del sistema

Establecer herramientas de software para el apoyo en la elaboración de diagramas técnicos

Crear una plantilla estandarizada de casos de uso basada en mejores prácticas de documentación

Definir formatos de aceptación para la validación sistemática de artefactos del sistema

1.3 Justificación

La elaboración de un modelo de dominio robusto es fundamental para el éxito de cualquier proyecto de software, particularmente en contextos complejos como el censo rural. Un modelo bien diseñado facilita la comunicación entre stakeholders, reduce la ambigüedad en los requisitos, y proporciona una base sólida para las fases posteriores de diseño e implementación (Evans, 2003).

La aplicación de estándares UML y metodologías ágiles como XP en el contexto de proyectos gubernamentales de gran escala contribuye a la transparencia, trazabilidad y calidad del software desarrollado, aspectos críticos cuando se manejan datos sensibles de ciudadanos (Sommerville, 2016).

2. Marco Metodológico

2.1 Metodología de Desarrollo: Extreme Programming (XP)

Para el proyecto "Censo Rural" se ha adoptado la metodología Extreme Programming (XP), una aproximación ágil que enfatiza la entrega frecuente de software funcional, la colaboración estrecha con el cliente, y la capacidad de respuesta ante cambios en los requisitos (Beck, 2000).

2.1.1 Justificación de la Selección de XP

La elección de XP como metodología de desarrollo se fundamenta en las siguientes características que se alinean con los requerimientos específicos del proyecto "Censo Rural":

Entregas Incrementales y Modulares: XP permite el desarrollo y entrega de módulos funcionales independientes, lo cual es crucial para un proyecto que requiere implementación por fases. En el contexto del censo rural, esto permite desarrollar primero el módulo de captura de datos (crítico para el trabajo de campo) y posteriormente integrar el módulo de análisis y generación de reportes.

Adaptabilidad a Cambios: La naturaleza dinámica de los proyectos gubernamentales y la necesidad de adaptarse a condiciones específicas de diferentes regiones rurales hacen que la flexibilidad inherente de XP sea una ventaja significativa.

Enfoque en el Usuario Final: XP prioriza la satisfacción del cliente y la entrega de valor real, aspectos fundamentales cuando se trabaja con encuestadores rurales y comunidades que requieren herramientas intuitivas y eficaces.

Pruebas Continuas: La metodología enfatiza las pruebas constantes y la integración continua, elementos esenciales para garantizar la funcionalidad del sistema en condiciones adversas y con conectividad limitada.

2.1.2 Impacto en la Arquitectura del Software

La adopción de XP dicta un enfoque arquitectónico modular o basado en componentes, donde cada módulo puede ser desarrollado, probado e implementado de manera independiente. Esta arquitectura facilita:

Desarrollo Paralelo: Diferentes equipos pueden trabajar simultáneamente en módulos distintos

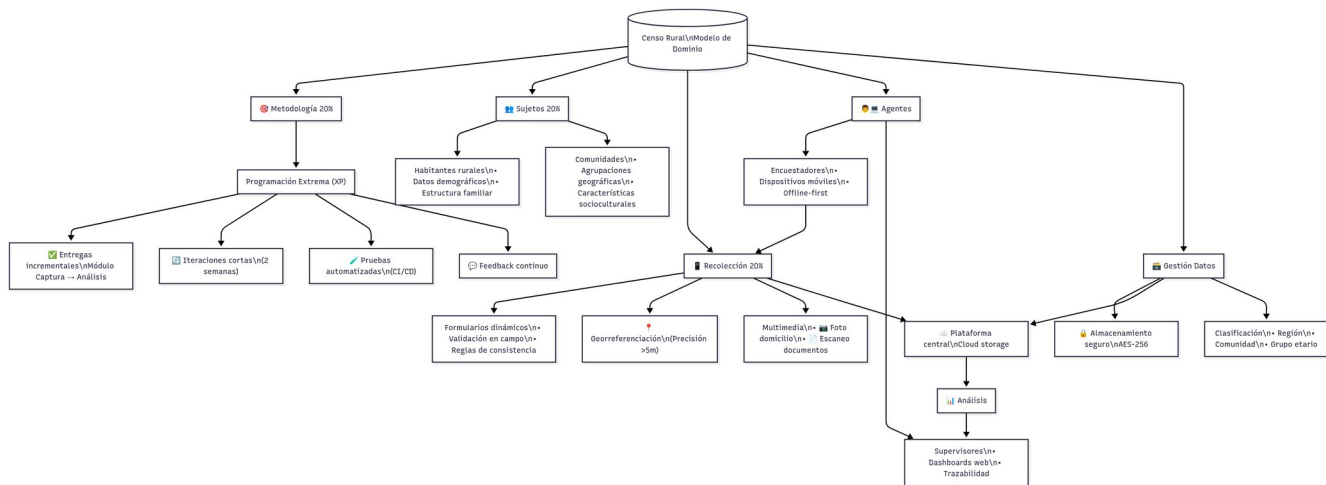
Pruebas Independientes: Cada módulo puede ser validado por separado antes de la integración

Despliegue Gradual: Los módulos pueden ser implementados progresivamente en diferentes regiones

Mantenimiento Simplificado: Los errores y mejoras pueden abordarse a nivel de módulo específico

2.2 Marco Teórico de Modelado de Dominio

El modelado de dominio se fundamenta en los principios del Domain-Driven Design (DDD) propuesto por Evans (2003), que establece que el software debe reflejar fielmente el dominio del problema que pretende resolver. Este enfoque implica:



Modelo Dominio (Debe contar con cuenta Gmail, es una extensión de GoogleDrive)

2.2.1 Lenguaje Ubicuo

Desarrollo de un vocabulario común compartido por todos los stakeholders del proyecto, desde desarrolladores hasta encuestadores rurales, que facilite la comunicación y reduzca malentendidos.

2.2.2 Modelo Rico en Conocimiento

Creación de un modelo que no solo capture datos sino también comportamientos y reglas de negocio específicas del dominio del censo rural.

2.2.3 Iteración Continua

Refinamiento constante del modelo basado en nueva información y feedback de usuarios finales.

3. Identificación del Tipo de Diagrama Apropriado

3.1 Análisis de Opciones de Modelado

Para el proyecto "Censo Rural", se han evaluado diferentes tipos de diagramas UML considerando las características específicas del dominio y los objetivos del sistema. La selección se basa en criterios de expresividad, claridad y utilidad para los stakeholders del proyecto.

3.1.1 Diagrama de Dominio

Descripción: El diagrama de dominio representa las entidades conceptuales del mundo real y sus

relaciones, sin considerar aspectos técnicos de implementación.

Justificación para el Proyecto: Es el tipo de diagrama más apropiado para modelar el dominio del "Censo Rural" porque:

Permite representar conceptos del mundo real como "Habitante", "Familia", "Comunidad Rural"

Facilita la comunicación con stakeholders no técnicos (funcionarios gubernamentales, líderes comunitarios)

Proporciona una base sólida para derivar otros artefactos técnicos

Capture la complejidad social y geográfica de las zonas rurales

3.1.2 Diagrama de Clases

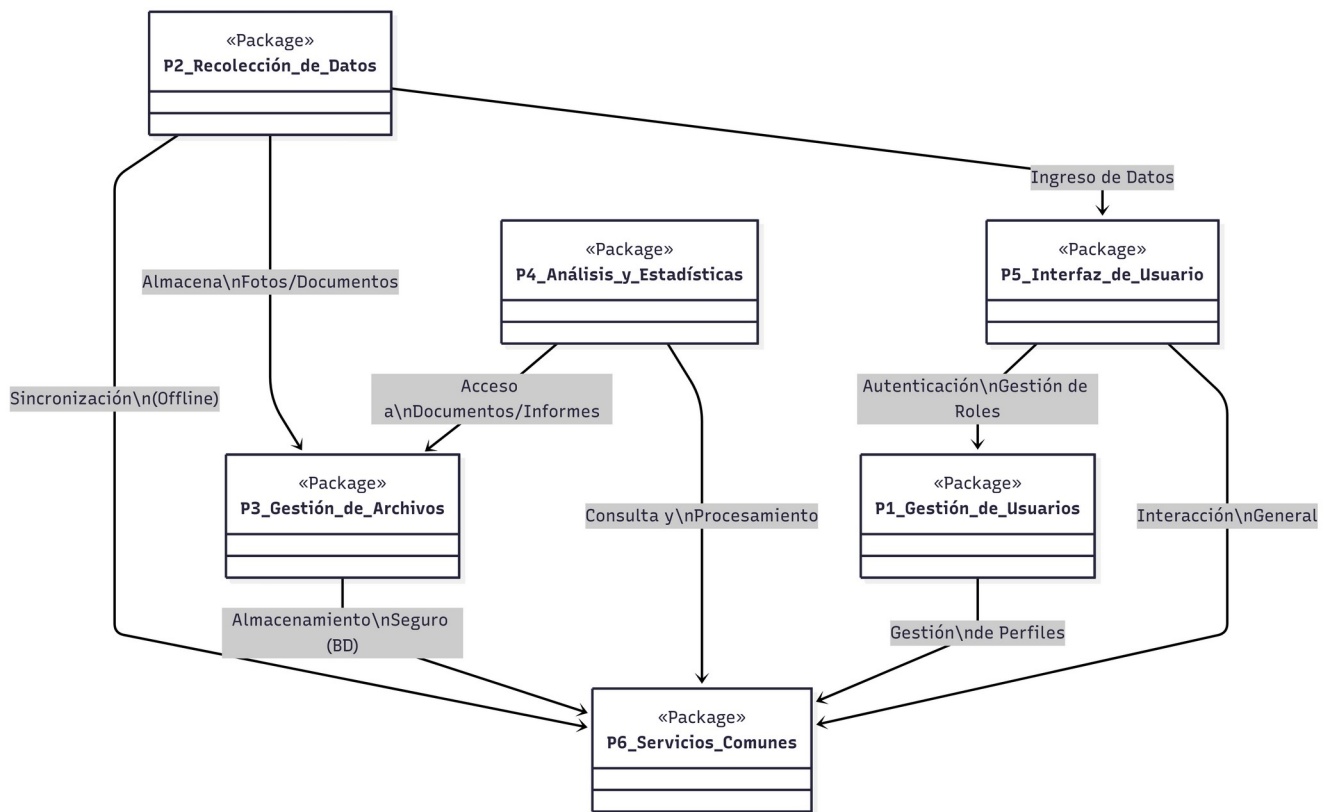
Descripción: Representa la estructura estática del sistema mostrando clases, atributos, métodos y relaciones.

Justificación: Complementa el diagrama de dominio proporcionando el detalle técnico necesario para la implementación del software.

3.1.3 Diagrama de Paquetes

Descripción: Organiza las clases en grupos lógicos y muestra las dependencias entre diferentes módulos del sistema.

Justificación: Fundamental para implementar la arquitectura modular requerida por la metodología XP.



[Diagrama de Paquetes](#) (Debe contar con cuenta Gmail, es una extensión de GoogleDrive)

3.2 Selección Final

Basado en el análisis anterior, se ha determinado que el Diagrama de Dominio es el tipo de diagrama más apropiado para el modelado inicial del proyecto "Censo Rural", complementado por diagramas de clases y paquetes para los aspectos técnicos de implementación.

Esta decisión se fundamenta en:

Complejidad del Dominio: El censo rural involucra múltiples actores (encuestadores, habitantes, supervisores) y procesos complejos que requieren representación conceptual clara

Diversidad de Stakeholders: El proyecto incluye stakeholders técnicos y no técnicos que necesitan un lenguaje común de comunicación

Base para Desarrollo: Proporciona los cimientos conceptuales para derivar otros artefactos técnicos necesarios

4. Modelado UML de Artefactos del Sistema

4.1 Diagrama de Clases

El diagrama de clases constituye uno de los artefactos centrales del sistema "Censo Rural", proporcionando una representación detallada de la estructura estática del software. Este diagrama traduce los conceptos identificados en el modelo de dominio a elementos técnicos implementables.

4.1.1 Metodología de Desarrollo

El desarrollo del diagrama de clases sigue un proceso iterativo que incluye:

Identificación de Clases: Derivación de clases a partir de sustantivos identificados en historias de usuario y casos de uso

Definición de Atributos: Especificación de propiedades que cada clase debe mantener

Identificación de Métodos: Determinación de operaciones que cada clase debe realizar

Establecimiento de Relaciones: Definición de asociaciones, herencias y dependencias entre clases

Refinamiento Iterativo: Ajustes basados en feedback y nuevos requisitos

4.1.2 Principios de Diseño Aplicados

Cohesión Alta: Cada clase tiene una responsabilidad específica y bien definida.

Acoplamiento Bajo: Las dependencias entre clases se minimizan para facilitar el mantenimiento y testing.

Principio de Responsabilidad Única: Cada clase tiene una sola razón para cambiar.

Encapsulación: Los detalles internos de implementación se ocultan mediante modificadores de acceso apropiados.

4.1.3 Espacio para Diagrama de Clases

[ESPACIO RESERVADO PARA INSERTAR EL DIAGRAMA DE CLASES DEL SISTEMA CENSO RURAL]

Descripción del Diagrama: El diagrama incluye las clases principales identificadas en el análisis del dominio, sus atributos, métodos y relaciones. Las clases centrales incluyen Usuario, Encuestador, Habitante, Encuesta, Formulario, DispositivoMovil, y SistemaAlmacenamiento, entre otras.

4.1.4 Clases Principales Identificadas

Clase Usuario (Abstracta)

Atributos: id, nombre, apellidos, documento, email, fechaCreacion

Métodos: autenticar(), actualizarPerfil(), cerrarSesion()

Descripción: Clase base que define la estructura común para todos los tipos de usuarios del sistema

Clase Encuestador (Hereda de Usuario)

Atributos: zonaAsignada, experiencia, dispositivoAsignado, encuestasRealizadas

Métodos: iniciarEncuesta(), sincronizarDatos(), validarFormulario()

Descripción: Representa a los funcionarios responsables de la recolección de datos en campo

Clase Habitante

Atributos: nombres, apellidos, documentoidentidad, edad, genero, nivelEducativo, ocupacion

Métodos: validarDatos(), obtenerInformacionBasica()

Descripción: Representa a los individuos objeto del censo rural

Clase Encuesta

Atributos: id, fechaRealizacion, coordenadas, estado, observaciones, encuestadorId

Métodos: inicializar(), guardarRespuesta(), finalizar(), sincronizar()

Descripción: Encapsula el proceso completo de recolección de datos para un hogar o individuo

4.2 Diagrama de Paquetes

El diagrama de paquetes organiza las clases del sistema en grupos lógicos, definiendo una arquitectura modular que facilita el desarrollo, mantenimiento y escalabilidad del sistema.

4.2.1 Arquitectura de Capas Propuesta

La organización de paquetes sigue el patrón arquitectónico de capas (Layered Architecture), que proporciona separación clara de responsabilidades y facilita el testing y mantenimiento.

4.2.2 Estructura de Paquetes

Capa de Presentación (Presentation Layer)

com.censorural.ui.activities: Actividades principales de la aplicación Android

com.censorural.ui.fragments: Componentes de interfaz reutilizables

com.censorural.ui.adapters: Adaptadores para listas y componentes complejos

com.censorural.ui.widgets: Controles personalizados para formularios

Capa de Aplicación (Application Layer)

com.censorural.services: Servicios de aplicación que orquestan casos de uso

com.censorural.controllers: Controladores que manejan la lógica de flujo

com.censorural.validators: Validadores de datos y reglas de negocio

Capa de Dominio (Domain Layer)

com.censorural.domain.entities: Entidades del dominio (Habitante, Encuesta, etc.)

com.censorural.domain.valueobjects: Objetos de valor (Coordenadas, Direccion)

com.censorural.domain.services: Servicios del dominio con lógica de negocio

Capa de Infraestructura (Infrastructure Layer)

com.censorural.data.local: Acceso a datos locales (SQLite)

com.censorural.data.remote: Comunicación con servicios remotos

com.censorural.data.repositories: Implementaciones de repositorios

com.censorural.utils: Utilidades y helpers

4.2.3 Espacio para Diagrama de Paquetes

[ESPACIO RESERVADO PARA INSERTAR EL DIAGRAMA DE PAQUETES DEL SISTEMA CENSO RURAL]

Descripción del Diagrama: El diagrama muestra la organización modular del sistema, las dependencias entre paquetes, y la separación clara de responsabilidades entre capas.

4.2.4 Beneficios de la Arquitectura Propuesta

Mantenibilidad: La separación en capas facilita la localización y corrección de errores.

Testabilidad: Cada capa puede ser probada independientemente mediante mocks y stubs.

Escalabilidad: Nuevas funcionalidades pueden agregarse sin afectar componentes existentes.

Reutilización: Los componentes de cada capa pueden reutilizarse en diferentes contextos.

5. Herramientas de Software Utilizadas

5.1 Selección de Herramientas

La selección de herramientas de software para el apoyo en la elaboración de diagramas se basa

en criterios de accesibilidad, compatibilidad con estándares UML, facilidad de uso, y capacidades de colaboración.

5.1.1 Draw.io (Diagrams.net)

Descripción: Herramienta de diagramación basada en web que ofrece capacidades completas para la creación de diagramas UML.

Justificación de Selección:

Accesibilidad: Disponible gratuitamente en línea sin necesidad de instalación

Compatibilidad: Soporte completo para estándares UML 2.5

Integración: Funciona con Google Drive, facilitando la colaboración y versionado

Versatilidad: Permite crear múltiples tipos de diagramas en una sola plataforma

Exportabilidad: Genera archivos en múltiples formatos (PNG, SVG, PDF, XML)

Funcionalidades Utilizadas:

Plantillas UML predefinidas

Biblioteca de formas estándar UML

Herramientas de alineación y distribución

Capas para organizar elementos complejos

Validación básica de sintaxis UML

5.1.2 Google Workspace

Google Drive:

Almacenamiento centralizado de diagramas y documentación

Control de versiones automático

Compartición controlada con stakeholders

Google Docs:

Documentación de especificaciones técnicas

Colaboración en tiempo real en documentos de requisitos

Comentarios y sugerencias para revisión

Google Sheets:

Matrices de trazabilidad

Listas de chequeo para validación

Seguimiento de progreso de artefactos

5.1.3 Herramientas Complementarias

Git/GitHub:

Control de versiones para código fuente de diagramas

Seguimiento de cambios en documentación técnica

Issues y milestones para gestión de tareas

Markdown Editors:

Documentación técnica en formato estándar

Facilita la generación de documentos web y PDF

Integración con sistemas de control de versiones

5.2 Flujo de Trabajo con Herramientas

5.2.1 Proceso de Creación de Diagramas

Análisis de Requisitos: Revisión de historias de usuario y casos de uso en Google Docs

Boceto Inicial: Creación de diagrama preliminar en Draw.io

Revisión Colaborativa: Compartición y comentarios a través de Google Drive

Refinamiento: Iteraciones basadas en feedback de stakeholders

Validación: Verificación contra listas de chequeo en Google Sheets

Versionado: Almacenamiento de versión final en repositorio Git

5.2.2 Estándares de Nomenclatura

Archivos de Diagramas:

Formato: PROYECTO_TIPO_VERSION.drawio

Ejemplo: CensoRural_DiagramaClases_v2.1.drawio

Documentos de Soporte:

Formato: PROYECTO_DOCUMENTO_FECHA.docx

Ejemplo: CensoRural_EspecificacionClases_20250803.docx

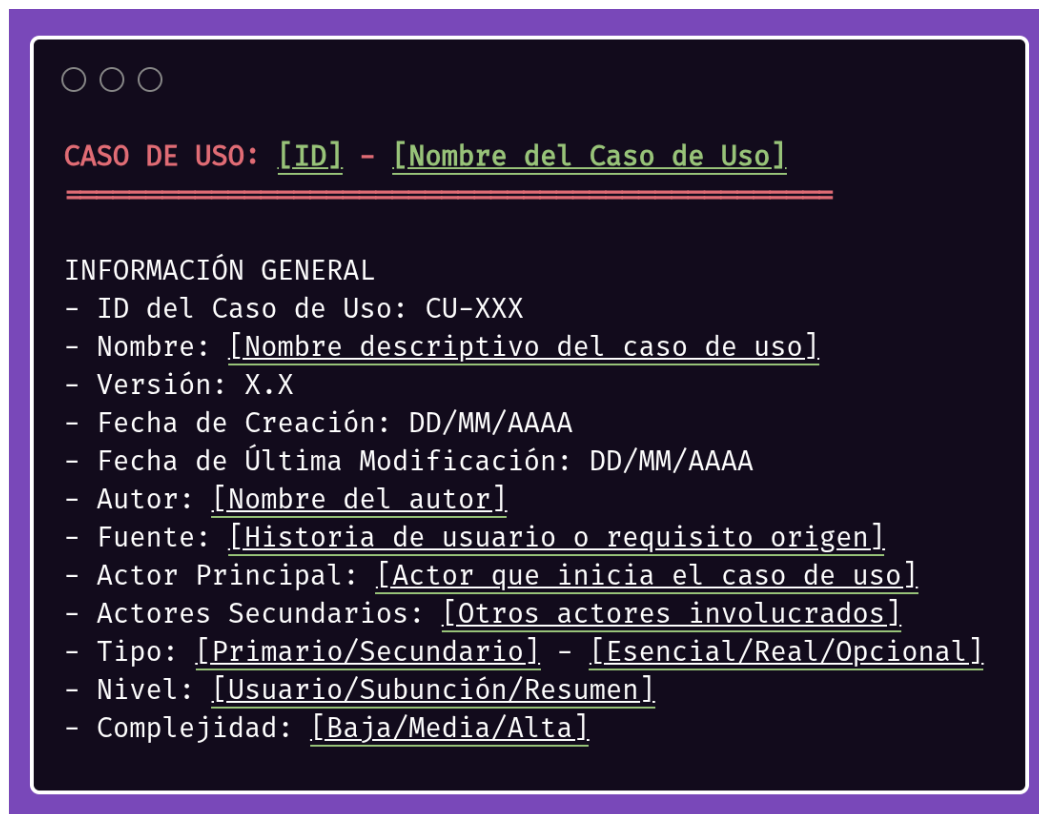
6. Plantilla de Casos de Uso

6.1 Estándares de Documentación

La plantilla de casos de uso desarrollada para el proyecto "Censo Rural" se basa en los estándares establecidos por Cockburn (2000) en "Writing Effective Use Cases" y las mejores prácticas del Rational Unified Process (RUP), adaptadas para el contexto específico de aplicaciones móviles en entornos rurales.

6.2 Estructura de la Plantilla

6.2.1 Encabezado del Caso de Uso



○ ○ ○

CASO DE USO: [ID] - [Nombre del Caso de Uso]

INFORMACIÓN GENERAL

- ID del Caso de Uso: CU-XXX
- Nombre: [Nombre descriptivo del caso de uso]
- Versión: X.X
- Fecha de Creación: DD/MM/AAAA
- Fecha de Última Modificación: DD/MM/AAAA
- Autor: [Nombre del autor]
- Fuente: [Historia de usuario o requisito origen]
- Actor Principal: [Actor que inicia el caso de uso]
- Actores Secundarios: [Otros actores involucrados]
- Tipo: [Primario/Secundario] - [Esencial/Real/Opcional]
- Nivel: [Usuario/Subunción/Resumen]
- Complejidad: [Baja/Media/Alta]

6.2.2 Descripción y Contexto



DESCRIPCIÓN

[Descripción concisa del propósito y objetivo del caso de uso, explicando qué se logra cuando se ejecuta exitosamente].

ALCANCE

[Definición clara de qué está incluido y qué está excluido del caso de uso].

NIVEL DE USUARIO

[Especificación del tipo de usuario y nivel de experiencia requerido para ejecutar el caso de uso].



ESCENARIO PRINCIPAL DE ÉXITO

1. [Primer paso del flujo principal]
2. [Segundo paso del flujo principal]
 - 2.1. [Sub-paso si es necesario]
 - 2.2. [Sub-paso si es necesario]
3. [Continuar numeración secuencial]
- ...
- N. [Último paso - caso de uso termina en éxito]

EXTENSIONES (FLUJOS ALTERNATIVOS)

[Numeración basada en el paso donde ocurre la desviación]

- 2a. [Condición alternativa en paso 2]:
 - 2a1. [Acción alternativa 1]
 - 2a2. [Acción alternativa 2]
 - 2a3. [Retorno al flujo principal o terminación]
- 3-5a. [Condición que puede ocurrir en pasos 3 a 5]:
 - 3-5a1. [Acción correspondiente]
 - 3-5a2. [Continuación o terminación]
- *a. [Extensión que puede ocurrir en cualquier momento]:
 - *a1. [Manejo de la situación excepcional]
 - *a2. [Acción de recuperación o terminación]

6.2.3 Precondiciones y Garantías



PRECONDICIONES

[Lista de condiciones que deben cumplirse antes de que el caso de uso pueda iniciarse]

1. [Precondición 1]
2. [Precondición 2]
- ...

POSTCONDICIONES (GARANTÍAS DE ÉXITO)

[Estado del sistema después de la ejecución exitosa]

1. [Postcondición 1]
2. [Postcondición 2]
- ...

GARANTÍAS MÍNIMAS

[Estado garantizado del sistema incluso si el caso de uso falla]

1. [Garantía mínima 1]
2. [Garantía mínima 2]
- ...

6.2.5 Requerimientos Especiales y Detalles Técnicos

6.3

Ejemplo Aplicado: Caso de Uso del Censo Rural

6.3.1 Caso de Uso Ejemplo: Captura de Datos Offline

```
○○○

flowchart TD
    %% 1. Acceso Encuestador
    subgraph Acceso["1. Acceso Encuestador"]
        direction TB
        A1[Campos de Entrada] --> A11["Nombre de Usuario\n(correo corporativo) [44,45]"]
        A1 --> A12["Contraseña [44,45]"]
        B1[Botones] --> B11["Iniciar Sesión"]
        B1 --> B12["¿Olvidó su contraseña? [44,46]"]
    end

    %% 2. Panel Principal
    subgraph Panel["2. Panel Principal"]
        direction TB
        A2["Bienvenido, [Nombre]!\nEstado: Online/Offline [7,48,49]"]
        B2[Formularios] --> B21["Censo Poblacional Rural 2025\n Descripción: Datos demográficos\n Progreso: 75% (Offline)"]
        B2 --> B22["Encuesta de Vivienda Rural\n Descripción: Infraestructura\n Progreso: 20% (Online)"]
        C2[Botones] --> C21["Sincronizar Datos [7,49]"]
        C2 --> C22["⚙ Configuración"]
        C2 --> C23["🚪 Cerrar Sesión [50]"]
    end

    %% 3. Captura Offline
    subgraph Captura["3. Captura Offline"]
        direction TB
        A3["Estado: Offline [7,48,49]"]
        B3[Datos] --> B31["Nombre Completo: _____"]
        B3 --> B32["Documento: _____"]
        B3 --> B33["Edad: _____"]
        B3 --> B34["Género: [ Masculino | Femenino | Otro ]"]
        C3[Ubicación] --> C31["📍 Capturar GPS [7,8,42]"]
        D3[Archivos] --> D31["📷 Tomar Foto [7,8,42]"]
        D3 --> D32["📎 Adjuntar Doc [7,8,42]"]
        E3[Botones] --> E31["💾 Guardar"]
        E3 --> E32["❌ Cancelar"]
    end

    %% Flujo principal
    Acceso -->|Credenciales válidas| Panel
    Panel -->|Selecciona formulario| Captura
    Captura -->|Guardar| Confirmación
    Confirmación -->|Nuevo registro| Captura
    Confirmación -->|Volver| Panel

    %% 4. Confirmación
    subgraph Confirmación["4. Confirmación"]
        direction TB
        A4["✅ Datos guardados!\n Registros pendientes: [X]\n Sincronización automática"]
        B4[Botones] --> B41["🌟 Nuevo Registro"]
        B4 --> B42["⬅ Volver al Panel"]
    end

    %% 5. Sincronización
    subgraph Sincro["5. Sincronización"]
        direction TB
        A5["Estado: Online\n 🔄 Sincronizando (2/5)\n ⌚ Tiempo: X segundos"]
        B5["✅ Sincronización completa"]
        C5[Botones] --> C51["📄 Ver Registros"]
        C5 --> C52["⬅ Volver al Panel"]
    end

    %% Conexiones adicionales
    Panel -->|Sincronizar| Sincro
    Sincro -->|Volver| Panel
```