

Especificación Formal de un Sistema de Votación Electrónica para el Contexto Peruano

Nombres Davis Yovanny Arapa Chua, Danny Quispe Cjuiro, Luis Gonzalo Basurco Monroy,
 Carlos Adrian Vizarreta Cheeya, Esther Mariana Chunga Pacheco
 Universidad La Salle

Introducción

La modernización electoral en el Perú requiere sistemas seguros y transparentes. Este trabajo modela formalmente el registro y verificación de votantes usando VDM++, garantizando integridad y consistencia en el padrón electoral.



Figura 1.Imagen Referencial 36. [1].

Objetivos

- Diseñar una especificación formal del módulo de registro y verificación de votantes.
- Asegurar integridad, trazabilidad y disponibilidad.
- Validar la identidad mediante DNI y biometría.
- Prevenir duplicaciones y estados inconsistentes.

Metodología

- Lenguaje: VDM++
- Herramienta: VDM Toolbox
- Clases modeladas

```

1 class Ciudadano
2 types
3   DNI = seq of char;
4   Nombre = seq of char;
5   Huella = seq of char;
6   EstadoRegistro = <Activo | Inactivo | Suspendido>;
7
8 CiudadanoInfo {
9   dni : DNI;
10  nombre : Nombre;
11  huellaDigital : Huella;
12  estado : EstadoRegistro;
13
14 instance variables
15  public datos : CiudadanoInfo;
16
17 inv len(datos.dni) >= 0;
18 inv forall x in set elem(datos.dni) & x in
19   set {'0','1','2','3','4','5','6','7','8','9'};
20 inv len(datos.nombre) > 0;
21 inv len(datos.huellaDigital) > 0;
22
23 operations
24  public Ciudadano : DNI * Nombre * Huella * EstadoRegistro ==> Ciudadano
25  Ciudadano(v_dni, v_nombre, v_huella, v_estado) == {
26    datos := m_CiudadanoInfo(v_dni, v_nombre, v_huella, v_estado);
27  }
28
29  public getDNI : () ==> DNI
30  getDNI() == return datos.dni;
31
32  public getNombre : () ==> Nombre
33  getNombre() == return datos.nombre;
34
35  public getHuella : () ==> Huella
36  getHuella() == return datos.huellaDigital;
37
38  public getEstado : () ==> EstadoRegistro
39  getEstado() == return datos.estado;
40
41 end Ciudadano

```

Figura 2. Especificación Formal en VDM++ - Clase Ciudadano. [2].

Diagrama de Clases

Estructura modular que integra ciudadanos, registro y validación.

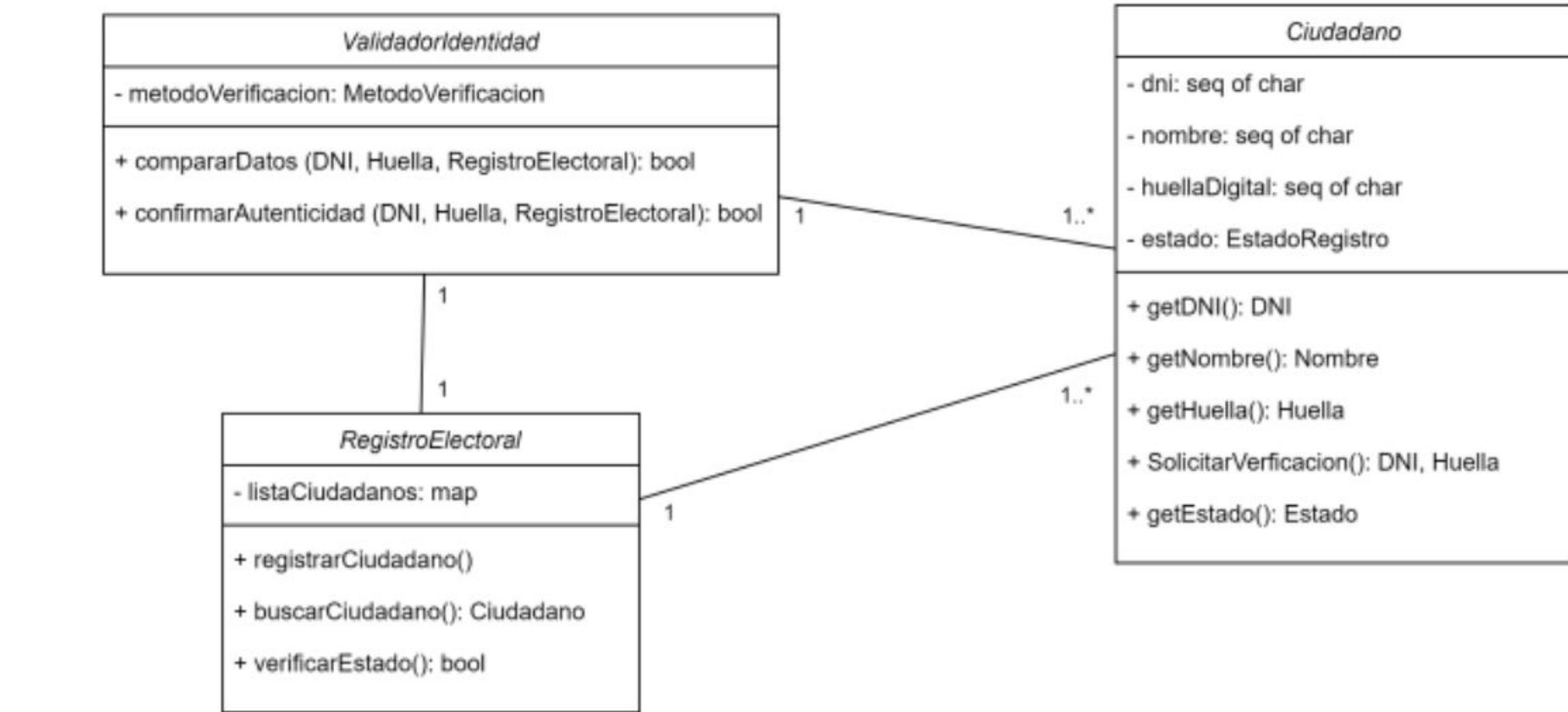


Figura 3. Diagrama de Clases. [3].

Resultados

- Validación al 100% de cobertura en todas las operaciones.
- Cumplimiento de requisitos funcionales (RF1–RF4).
- Modelo consistente y listo para integración en un SVE real.

Referencias

1. Alasmay et al. (2023) – Blockchain y biometría.
2. Berenjestanaki et al. (2024) – ZK-SNARKs.
3. Ibarra et al. (2023) – Verificación multiagente.
4. Fiore et al. (2024) – Claves desde biometría.