

Documento del
modelo de
Arquitectura de
Sistema del
voto electrónico
en el municipio
de Choachí

October 10

2011

DOCUMENTO
DE
ARQUITECTURA
SOFTWARE

Tabla de Contenido

1.	Introducción	2
2.	Referencias	3
3.	Características generales del software	3
4.	Requisitos atendidos por el diseño	4
5.	Identificación y selección de patrones de negocio	7
6.	Diseño de software de alto nivel	10
6.1.	Introducción al patrón de arquitectura N-Tier	11
7.	Identificación y selección de patrones de diseño	11
8.	Validación de la arquitectura de software	28
8.1.	Presentación de la arquitectura	29
8.2.	Presentación de los factores de negocio	29
8.3.	Análisis de resultados	35

Documento de Arquitectura de Software

1. Introducción

El proceso de diseño de unas aplicaciones para dar vida al voto electrónico en el municipio de Choachí no puede ser un proceso improvisado y mucho menos empírico. Es por eso, que como elemento innovador e integrador se vio necesario reconocer previamente la estructura de la línea estratégica del voto en la Registraduría municipal de Choachí para reconocer cada uno de los procesos que componen esta línea, su funcionamiento e identificar los puntos débiles o de mejora que se podían trabajar dentro de esta línea estrategia.

Este elemento lo trabajamos como la Arquitectura empresarial de la Registraduría municipal de Choachí (documento anterior), en la cual se identificaron todos los procesos propios del voto, además de pensarse y posteriormente establecerse en la vista de *comportamiento de la aplicación* una serie de aplicaciones necesarias para soportar las mejoras en los procesos como son: *MORPHOCHECK*, *registro de candidatos*, *registro de electores*, *votación*, y *escrutinio de resultados*. Cabe resaltar que, de estas aplicaciones se descarta la creación de *MORPHOCHECK* ya que es un producto adquirido por la Registraduría Nacional del Estado Civil, la cual en el momento de implementación de este modelo de Arquitectura de Sistema, le podría solicitar a la empresa SAGEM (creadora de *MORPHOCHECK*) diseñar una interface para integrar los datos que se requieran de esta aplicación para el resto del modelo de arquitectura de software para el voto electrónico en el municipio de Choachí.

El documento está organizado alrededor de tres ideas principales.

- Las características generales del diseño
- Los requisitos atendidos por el diseño
- Los modelos y vistas que lo detallan

Al contrario de muchas otras actividades técnicas, el desarrollo de sistemas intensivos en software dedica la mayoría de sus esfuerzos a la especificación y modelado.

Los modelos son utilizados tanto para el análisis de requisitos, como para el diseño de la solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación.

Los modelos son presentados por vistas o diagramas, generalmente utilizando notaciones gráficas como el UML.

2. Referencias

Las referencias aplicables a este documento son:

- IEEE 1471-2000 ST
- Technology Acceptance Model
- ISO 9126 -2001 Calidad del Software y Métricas de evaluación
- The 4+1 View .Kruchten – 1009

3. Características generales del software

El proceso de construir la arquitectura de software para el voto electrónico en el municipio de Choachí está fundamentado en la necesidad de una herramienta de apoyo para mejorar los procesos propios de la línea estratégica de voto de la registraduría municipal de Choachí.

Según esto, se puede apreciar claramente la relación que hay entre los procesos de negocio y la aplicación que requeriría. Entendiendo para este documento que el termino aplicación que se usa en arquitectura empresarial se refiere en alto nivel a una funcionalidad del sistema que se construya. La siguiente tabla mapea como está cada proceso de negocio reflejado en las aplicaciones (funcionalidades) a construir.

	Proceso de inscripción de candidato	Proceso inscripción de electores	Proceso cierre de candidatos	Proceso cierre de electores	Proceso de votación	Proceso de escrutinio de resultados
Aplicación Registro candidato	X		X			
Aplicación Registro elector		X		X		
Aplicación votación					X	
Aplicación resultado escrutinio						X

Tabla 1. Procesos de negocio versus aplicaciones a diseñar. Autor: Daniel Cáceres

De allí, que como elemento esencial para fundamentar la construcción de software, recurramos a la vista de comportamiento de la aplicación desarrollada en el documento de Arquitectura Empresarial (application behavior viewpoint).

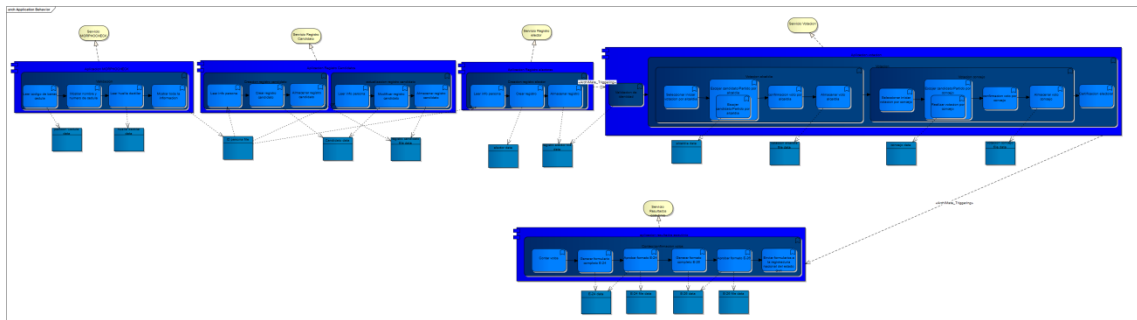


Figura 1. Vista de comportamiento de la aplicación

La figura 1 completa la puede ver en el documento de Arquitectura empresarial sección 14.

Aquí apreciamos que desde la Arquitectura Empresarial se pre-diseña un software que tenga 5 aplicaciones: *MORPHOCHECK*, *registro candidatos*, *registro electores*, *votación*, y *resultados escrutinio*, descartando la creación de la aplicación MORPHOCHECK por ser un producto ya adquirido por parte de la Registraduría Nacional del Estado Civil a la empresa SAGEM.

4. Requisitos atendidos por el diseño

Basado en la vista de comportamiento de la aplicación, se hace un proceso de valoración del estado actual de la registraduría al estado deseado mediante el proceso de diseño de casos de usos, para poder apreciar lo que se necesita, como se necesita y quienes interactuarían con la herramienta a desarrollar.

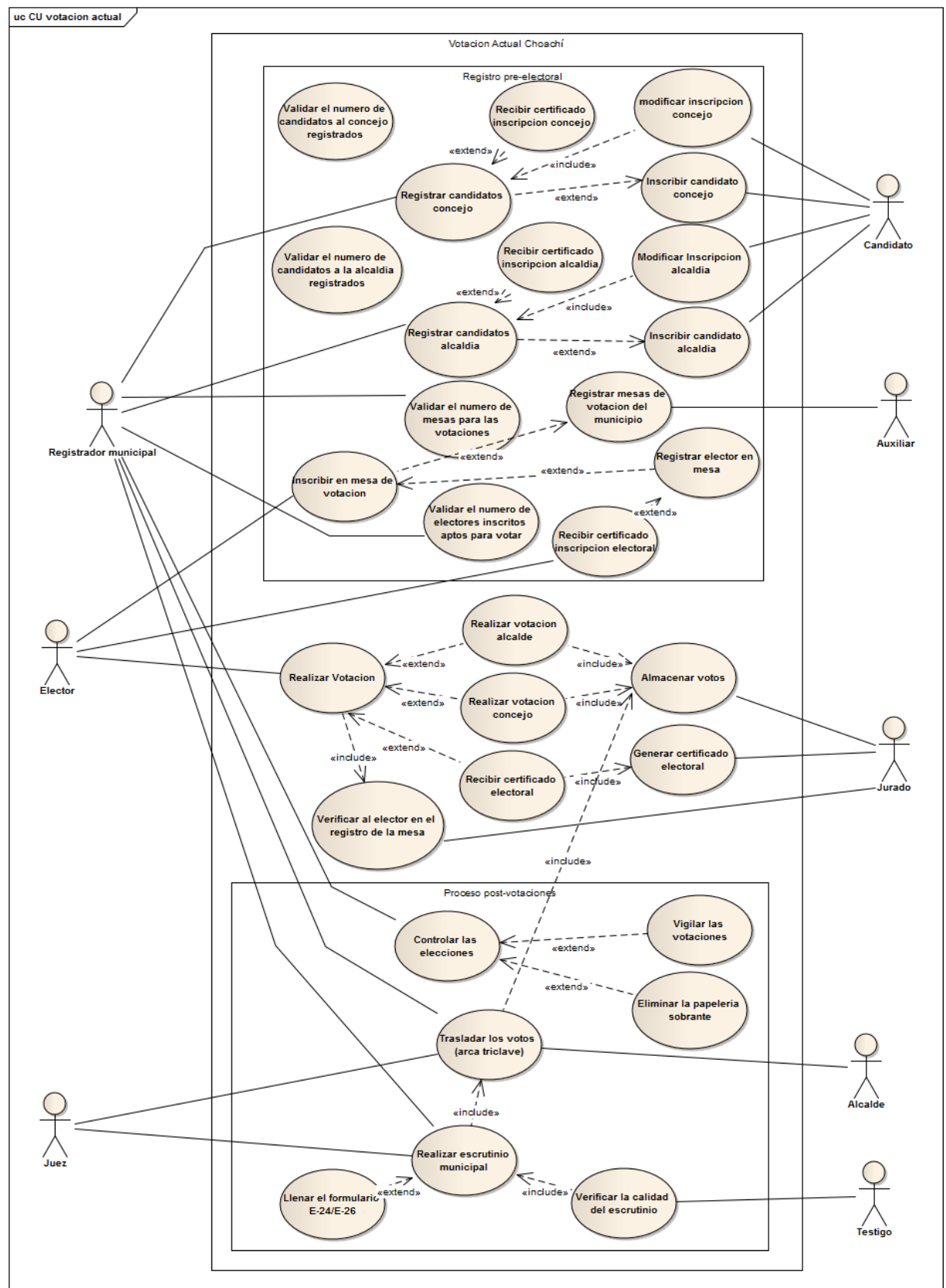


Figura 2. Estado actual de la registraduría municipal de Choachí con respecto al voto

En el caso de uso actual, se puede apreciar que todos los procesos que se llevan a cabo en Choachí son de forma manual, no hay ninguno con soporte tecnológico o automatizado, e influyen muchos actores.

Basado en el estado actual, se plantea un estado deseado para el voto, donde ya no será el voto de forma manual sino se busca obtener el voto electrónico para el municipio.

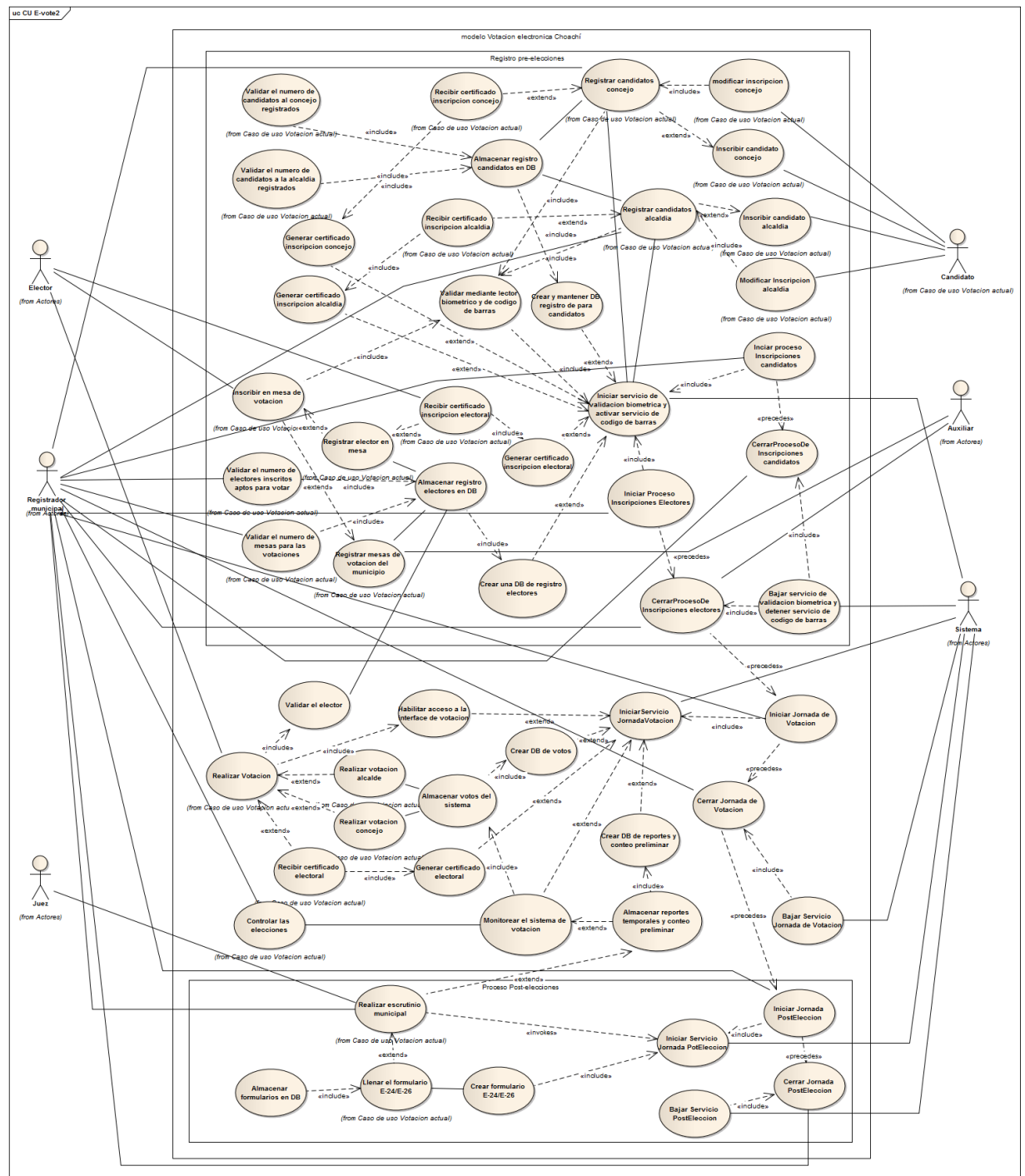


Figura 3. Estado deseado del voto electrónico en el municipio de Choachí

Para el estado deseado del voto electrónico, se espera mejorar la seguridad de los procesos que constituyen esta línea estratégica y otorgarle nuevas formas de auditabilidad a los mismos en cualquier momento. Además de intentar mejorar el funcionamiento de estos procesos mediante la automatización o implementación de herramientas tecnológicas para cumplir con el voto.

5. Identificación y selección de patrones de negocio

Con base en la arquitectura empresarial definida anteriormente, se puede establecer según la metodología BPTrends [1], una relación de los procesos que tiene la Registraduría municipal de Choachí contra el uso los mismos como procesos intensivos definidos como procesos intensivos en sistemas, procesos intensivos en personas, procesos intensivos en decisión y procesos intensivos en documentos.

	Procesos intensivos en sistemas	Procesos intensivos en personas	Procesos intensivos en decisión	Procesos intensivos en documentos
Proceso de inscripción de candidato	X	X		
Proceso inscripción de electores	X	X		
Proceso cierre de candidatos	X		X	
Proceso cierre de electores	X		X	
Proceso de votación	X	X		
Proceso de escrutinio de resultados	X	X		

Tabla 2. Procesos de la registraduría municipal de Choachí vs procesos según BPTrends. Autor: Daniel Cáceres

Este mapeo entre procesos y procesos intensivos sirve como elemento de identificación previo para poder establecer que si sea necesario automatizar algunos procesos o reorganizarlos. Según se aprecia en la tabla 2, se ve que los procesos de la registraduría si pueden ser aptos para ser automatizados lo que a su vez trae el reorganizar algunos.

Teniendo claro que según el análisis de BPTrends la arquitectura empresarial (específicamente la vista de comportamiento de la aplicación) tiene toda la

viabilidad, se debe buscar una forma de relacionar ese modelo a un diseño de arquitectura de software.

La mejor forma de hacerlo, es mediante la identificación de patrones de negocio (los cuales darán mayor fortaleza a la arquitectura empresarial) que posteriormente puedan ser mapeados a patrones de software (tanto de arquitectura como de diseño) ligándola directamente a la arquitectura empresarial con la arquitectura de software.

Forster plantea en [2] unos patrones de mejora de los procesos de negocio que se clasifican en 4 tipos: Patrones básicos de control de flujo, Patrones avanzados de ramificación y sincronización, patrones estructurales y patrones de múltiples instancias. Según Forster, describe estos patrones de mejora de procesos de negocio como un intento para describir soluciones satisfactorias para modelar los pasos de mejora operacionales de un proceso de negocio.

Basado en este concepto, los procesos de negocio del voto electrónico fueron mapeados a estos patrones, identificándolos mediante el subproceso perteneciente a cada proceso del modelo de arquitectura empresarial de mayor influencia para el diseño según la definición de cada patrón de negocio.

	Patrones básicos de control de flujo	Patrones avanzados de ramificación y sincronización	patrones estructurales	patrones de múltiples instancias
Proceso de inscripción de candidato			Morphocheck	
Proceso inscripción de electores			Morphocheck	
Proceso cierre de candidatos	Omitir validación usuario (registrador)			
Proceso cierre de electores	Omitir validación usuario (registrador)			
Proceso de votación			Morphocheck	
Proceso de escrutinio de resultados	Omitir publicación resultados físicos			

Tabla 3. Patrones de negocio de Forster versus Procesos de negocio de la Registraduría municipal de Choachí. Autor: Daniel Cáceres.

Según el mapeo, los patrones de negocio que reflejan las necesidades de los procesos definidos en la arquitectura empresarial por mayor cantidad de procesos relacionados son los Patrones básicos de control de flujo y los patrones estructurales.

El problema que trae estos patrones, es que para llevarlos a un nivel más bajo hay que estar ligado obligatoriamente al Framework de Forster. Dada esta situación, se buscó unos patrones de negocio más específicos, situación que nos llevó a encontrar en la base del conocimiento información de Kim et. Al. Según *Kim et al.* en [3] comentan que los patrones de Forster pueden ser ajustados a un nivel de detalle más concreto sin emplear su framework. Lo primero que plantean *Kim et al* es que se le cambie el término de patrones de mejora de procesos de negocio por patrones de cambio de procesos de negocio, y luego definen 3 nuevas categorías con base a los patrones definidos anteriormente por Forster y los patrones asociados a cada categoría: Patrones de extensión/borrado de actividad, Patrones de fusión/cambio de actividad, y patrones de división/cambio de actividad.

Dentro del documento de Kim et al, no presenta una relación directa con los patrones de Forster, solo dice que los tomo como base. Para este documento, el autor realiza una relación de los patrones de Forster con los de Kim et al basándose en los conceptos y términos definidos en sus artículos previamente citados.

PATRONES DE FORSTER	PATRONES DE KIM ET AL
Patrones básicos de control de flujo	Patrones de extensión/borrado de actividad
Patrones avanzados de ramificación y sincronización	Patrones de fusión/cambio de actividad
patrones estructurales	Patrones de fusión/cambio de actividad
patrones de múltiples instancia	patrones de división/cambio de actividad

Tabla 4. Patrones de negocio de Forster versus Procesos de negocio de la Registraduría municipal de Choachí. Autor: Daniel Cáceres

El resultado de esta tabla refleja que los procesos de negocio de la Registraduría municipal de Choachí con respecto a la línea estrategia del voto según el esquema de Kim et al, estarían reflejados en los patrones de extensión/borrado y fusión/cambio de actividad. Basado en esta comparación, para ir a un nivel de detalle más específico y no mantener la generalidad que representaba los patrones de Forster, se presenta la siguiente tabla donde se

mapearan los procesos de negocio de la registraduría versus los patrones de Kim et. Al, identificando el patrón específico de negocio de cada clasificación con el respectivo proceso de negocio.

	Patrones de extensión/borrado de actividad	Patrones de fusión/cambio de actividad	Patrones de división/cambio de actividad.
Proceso de inscripción de candidato		Patrón de cambio de contenido de actividad	
Proceso inscripción de electores		Patrón de cambio de contenido de actividad	
Proceso cierre de candidatos	Patrón de borrado/desvió de actividad		
Proceso cierre de electores	Patrón de borrado/desvió de actividad		
Proceso de votación		Patrón de cambio de contenido de actividad	
Proceso de escrutinio de resultados	Patrón de borrado/desvió de actividad		

Tabla 5. Patrones de negocio de Kim et al. versus Procesos de negocio de la Registraduría municipal de Choachí. Autor: Daniel Cáceres.

6. Diseño de software de alto nivel

Debido a que los patrones de negocio identificados son el Patrón de *cambio de contenido de actividad* y el patrón de borrado/desvió de actividad, pertenecientes a dos categorías de clasificación de patrones diferentes como lo son el "*fusión/cambio de actividad*" y el "*extensión/borrado de actividad*", se buscó dentro de la base del conocimiento la forma de asociarlos al diseño del software. Vistos estos patrones y debido a la diferencia entre ellos dada su clasificación, se encontró que la implementación de estos patrones de negocio en software sólo se puede hacer mediante un patrón arquitectónico definido como el patrón layers [4] o el Layered architecture pattern [5].

Considerando que el patrón layers su objetivo es descomponer en subtareas la estructura del software donde cada subtask tiene una función especificada, se

utilizará el patrón N-Tier [6] como la aproximación más clara de este patrón para su implementación en un web service.

6.1. Introducción al patrón de arquitectura N-Tier

El patrón N-Tier es una generalización de la arquitectura de tres capas. La arquitectura de 3 capas consiste en una capa de presentación, una de negocio y una de bases de datos y la lógica asociada es ésta, cuando se intenta incorporar en este arquitectura de 3 capas la parte de los servicios web se presentan problemas de flexibilidad con la arquitectura. Esto hace que se requiera incorporar una capa adicional de tal suerte que surge la arquitectura N-Ter como se muestra continuación:



Figura 4. Diseño de alto nivel. Patrón arquitectónico general n-tier para el voto electrónico.

Ya con un patrón de arquitectura definido para el software, podemos empezar a ir a un nivel de más detalle.

7. Identificación y selección de patrones de diseño

Hay muchas formas para poder usar e interpretar los casos de uso (y sus respectivos requerimientos) para llegar a su implementación. Una de las formas que tiene mayor solicitud por la calidad en que se desarrolla la solución es mediante el mapeo de estos casos de uso a atributos de calidad. Pero, ¿a qué nos referimos con calidad? Definiciones hay muchas, pero la que mayor representación tiene en el mundo de la ingeniería de sistemas (en especial la ingeniería del software) es la definición que da la IEEE en el Standard 729-1983:

- *“La totalidad de los rasgos y las características de un producto de software que le confieren la capacidad de satisfacer determinadas necesidades: por ejemplo, cumplimiento de las especificaciones.*
- *El grado en que el software posee una combinación deseada de atributos.*
- *El grado en que un cliente o un usuario percibe que el software cumple con sus expectativas compuesto.*
- *Las características de composición de un software que determinan el grado en que el software en uso complacerá las expectativas del cliente”.*

De estos conceptos de la IEEE, el enfoque se hará sobre la segunda definición, donde se habla de la combinación deseada de atributos. Basados en este concepto, deducimos que los atributos de calidad son características que tienen relación directa con el lenguaje de programación y el entorno para el cual se va a implementar el producto software.

Distintos modelos de los atributos de calidad han sido inventados como el modelo de McCall [7], el modelo de Boehm [8], el modelo FURPS [9], el estándar ISO/IEC 9126 [10], el modelo Dromey [11], el modelo de estrella [12], el modelo de redes bayesianas [13] y el modelo *“nutshell”* de Khosravi y Guéhéneuc [14] que será el modelo que usaremos para nuestros atributos de calidad.

Pero reconocer estos atributos de calidad no significa que el desarrollo ya sea de calidad. De allí, que diversos autores buscaran la forma de crear algún tipo de estándar en el desarrollo software para mejorar la producción del mismo, y uno de esos autores fue Eric Gamma y su grupo quienes crearon los “Patrones de diseño” [15].

Posteriormente, diversos autores generaron vínculos entre los atributos de calidad y los patrones de diseño. Para este desarrollo se tomó en cuenta la tabla de comparación de Khosravi y Guéhéneuc sin los atributos de independencia del software ni independencia del hardware, donde dependiendo de la calificación de todos los atributos del modelo “Nutshell” se le asignaba un patrón de diseño.

Design Patterns	Quality Sub-characteristics and Characteristics								
	Expendability	Simplicity	Generality	Modularity	Learnability	Understandability	Operability	Scalability	Robustness
Abs. Fact.	E	E	G	G	G	G	G	G	G
Builder	G	G	F	F	F	G	F	G	G
Fact. Met.	P	P	F	G	G	G	G	G	G
Prototype	E	G	F	G	F	G	F	E	G
Singleton	P	B	F	E	F	F	F	G	G
Adapter	F	F	P	G	G	F	F	G	F
Bridge	G	F	G	G	F	F	G	G	G
Composite	F	F	F	F	F	G	F	F	G
Decotator	E	E	G	F	G	G	G	G	F
Façade	G	G	G	G	F	G	F	F	F
Flyweight	P	P	F	G	G	P	F	G	G
Proxy	G	P	F	G	F	P	G	G	F
Chain of Res.	G	G	G	P	F	F	G	P	F
Command	G	P	F	F	P	B	G	G	G
Interpreter	G	F	G	F	F	F	G	G	F
Iterator	E	E	G	F	G	F	F	G	G
Mediator	G	F	G	G	F	F	G	G	F
Memento	G	F	F	B	P	F	G	F	P
Observer	E	G	E	F	F	G	G	G	G
State	G	G	F	P	F	B	G	G	F
Strategy	G	F	P	F	P	P	F	P	F
Tem. Met.	E	G	F	F	G	G	G	G	G
Visitor	E	G	G	F	G	P	F	G	F

Excellent
Good
Fair
Bad
Poor

Figura 5. Tabla de comparación atributos de calidad versus patrones de diseño. Autor: Khosravi y Guéhéneuc, modificada por: Anónimo.

Basado en estos conceptos, se reconocieron los atributos de calidad del software para voto electrónico a desarrollar y se les asignó una calificación. Para ser más específico, se calificaron los atributos de calidad por cada aplicación para poder mirar que tipo de patrón se puede identificar como esencial por aplicación. La clasificación se hace con los parámetros de Excellent (E), Good (G), Fair (F), Bad (B), y Poor (P) que define Khosravi y Guéhéneuc.

La asignación de calificación por atributo de calidad se hace con base a la información obtenida mediante la entrevista con el Registrador municipal, junto con el conocimiento adquirido mediante la base del conocimiento en la interpretación de las definiciones de los atributos de calidad que aglomera Khosravi y Guéhéneuc en [14].

	Registro candidato	Registro electores	Voto	Resultados
Usabilidad (6)	E	E	E	E
Formación (5)	E	E	E	E
Simplicidad (2)	E	E	E	E
Tolerancia al error (9)	G	G	E	E
Capacidad de expansión (8)	G	F	E	E
Generalidad (3)	G	G	G	G
Modularidad (4)	E	E	E	E
Operatividad (7)	E	E	E	E
Expendability (1)	E	E	E	E

Tabla 6. Aplicaciones versus atributos de calidad. Autor: Daniel Cáceres.

Después de haber calificados los atributos, se busca cual patrón de diseño se puede acoplar. Para este proyecto se trabajará con patrones base por aplicación a pesar de las múltiples coincidencias entre características y atributos de las cuatro aplicaciones a trabajar (registro candidato, registro electores, voto, resultado).

	1	2	3	4	5	6	7	8	9
Registro candidato	E	E	G	E	E	E	E	G	G
Registro electores	E	E	G	E	E	E	E	F	G
Voto	E	E	G	E	E	E	E	E	E
Resultado	E	E	G	E	E	E	E	E	E

Tabla 7. Aplicaciones versus Atributos de calidad según Khosravi y Guéhéneuc. Autor: Daniel Cáceres

Según esto, no hay un patrón que acierte en su totalidad con cada aplicación. Los parecidos son:

Registro candidato	E	E	G	E	E	E	E	G	G	
Patrón										probabilidad de acierto
abstract factory	E	E	G	G	G	G	G	G	G	0.55555556
Decorator	E	E	G	F	G	G	G	G	F	0.44444444
Iterator	E	E	G	F	G	F	F	G	G	0.55555556

Tabla 8. Aplicación Registro candidato versus patrones de diseño. Autor: Daniel Cáceres

El patrón para la aplicación de registro de candidatos que se va a utilizar en el diseño es Abstract Factory debido a que es un patrón de creación que abstrae el tipo concreto de objeto producido es decir no cambia si el objeto a crear es de otro tipo. En nuestro caso las características del candidato puedan cambiar pero sin importar estos cambios su forma de interpretación será la misma. No

obstante se ve reflejado el patrón Iterator en la aplicación debido a que ésta también debe soportar el proceso de cierre de candidatos en el cual se bloquea el ingreso de nuevos registros y se genera un conteo de la totalidad de los candidatos inscritos.

Registro electores	E	E	G	E	E	E	E	F	G	
Patrón										probabilidad de acierto
abstract factory	E	E	G	G	G	G	G	G	G	0.44444444
Decorator	E	E	G	F	G	G	G	G	F	0.33333333
Iterator	E	E	G	F	G	F	F	G	G	0.44444444

Tabla 9. Aplicación Registro elector versus patrones de diseño. Autor: Daniel Cáceres

El registro de electores es abstract Factory debido a la similitud de este proceso con el proceso de registro candidatos, salvo que los registros de los electores no pueden modificarse.

Aplicación votación	E	E	G	E	E	E	E	E	E	
Patrón										probabilidad de acierto
abstract factory	E	E	G	G	G	G	G	G	G	0.33333333
Decorator	E	E	G	F	G	G	G	G	F	0.33333333
Iterator	E	E	G	F	G	F	F	G	G	0.55555556

Tabla 10. Aplicación votación versus patrones de diseño. Autor: Daniel Cáceres

El patrón que se escoge para el proceso de votación es el Iterator debido a que este patrón facilita el recorrido de la selección de opción por candidato, es decir permitirá que el elector pueda fácilmente anular o devolver una opción seleccionada ya sea por alcaldía o por concejo.

Resultados escrutinio	E	E	G	E	E	E	E	E	E	
Patrón										probabilidad de acierto
abstract factory	E	E	G	G	G	G	G	G	G	0.33333333
Decorator	E	E	G	F	G	G	G	G	F	0.33333333
Iterator	E	E	G	F	G	F	F	G	G	0.55555556

Tabla 11. Aplicación Resultados escrutinio versus patrones de diseño. Autor: Daniel Cáceres

Por cada uno de los candidatos como por partido se deben hacer recorridos sobre los registros almacenados para generar los resultados de la votación, es por esto que el patrón Iterator es quien permite realizar ésta navegación.

Definidos los patrones de diseño por aplicación, hay que hacer una aproximación de estos a los patrones para Web Services debido a que el diseño de este sistema de voto electrónico debe ser diseñado y en su momento implementado para Web Services debido a las políticas nacionales de sitios web definidos dentro de la Registraduría Nacional del Estado Civil y en el programa Gobierno en línea.

Considerando esta restricción y entendiendo que los patrones de diseño pueden ser utilizados en lenguajes de programación como C++, C# y Java con sus respectivos cambios ajustados al lenguaje, se escoge para este proyecto diseñar la arquitectura de software con base en el lenguaje Java debido a la fortaleza en seguridad y las estructuras a nivel de patrones que para Web Services ofrece Java [6][16][17], situación que no ofrece Microsoft; esto complementado con la especificación de propiedad intelectual que se hizo de la propuesta completa donde se especificó el uso de software libre con licencia GPL versión 3, aceptada tanto por el director de la tesis como por el evaluador de la propuesta.

Según Paul Monday [6], los patrones que se acercan al Abstract Factory y al Iterator de los patrones GOF son: El patrón Service Factory y el Business Process (Composition).

El patrón Service Factory es un patrón que tiene sus raíces el patrón GOF Abstract Factory. La idea de este patrón es simple: debe aislar los puntos de la variabilidad en bloques de código de contenido fácilmente. Es decir, el uso de este patrón es para abstraer su aplicación fuera de los detalles de escoger un servicio web socio. La responsabilidad del Service Factory es seleccionar una de las implementaciones de servicio y retornar una implementación trabajando que se adhiera a la interface común de la aplicación.

El patrón Business Process (Composition) es un patrón cuya intención es ilustrar y proveer una guía para combinar las actividades de negocio en un único y consumible Web Service con una interface bien definida. Este patrón también es un intento para cambiar la mentalidad de la perspectiva orientada a objetos del lenguaje java a una perspectiva de procesos de negocio.

Siguiendo estos conceptos, la relación de las aplicaciones a los patrones de diseño y luego a los patrones de Web Services sería:

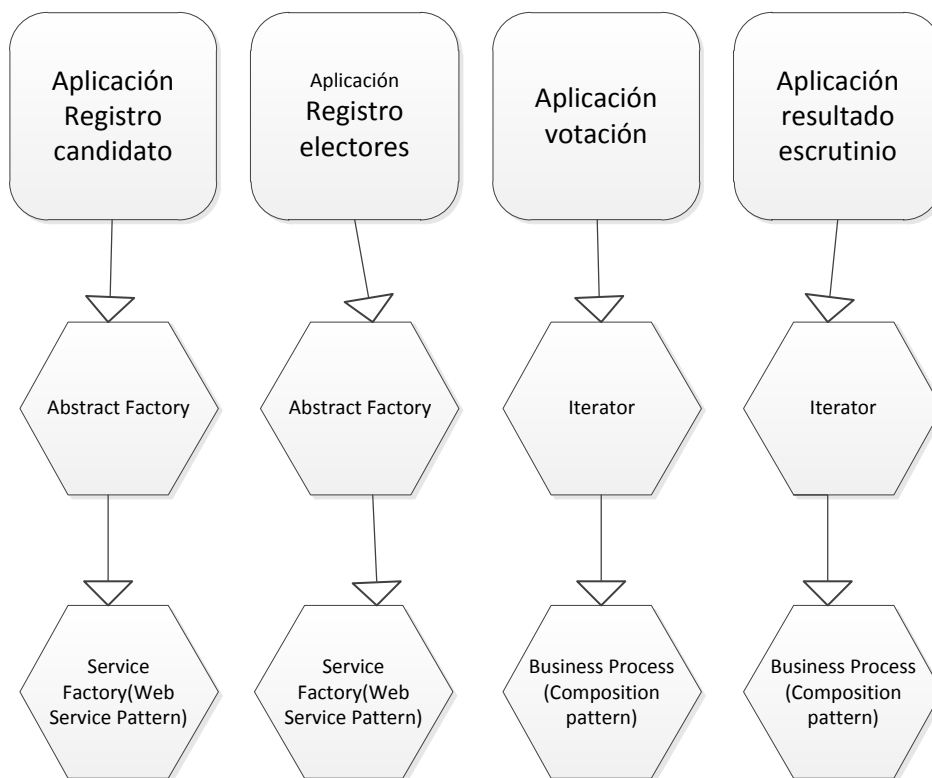


Figura 6. Mapeo de aplicaciones a patrones GOF y a Patrones de Web Services.

Ahora, se puede apreciar que las aplicaciones de registro candidato y registro electores comparten el mismo patrón GOF como de Web Services, de igual forma la aplicación Votación con la aplicación Resultado Escrutinio. Con base a esto, se integraran estos patrones de web services al patrón arquitectónico definido anteriormente q es el N-Tier.

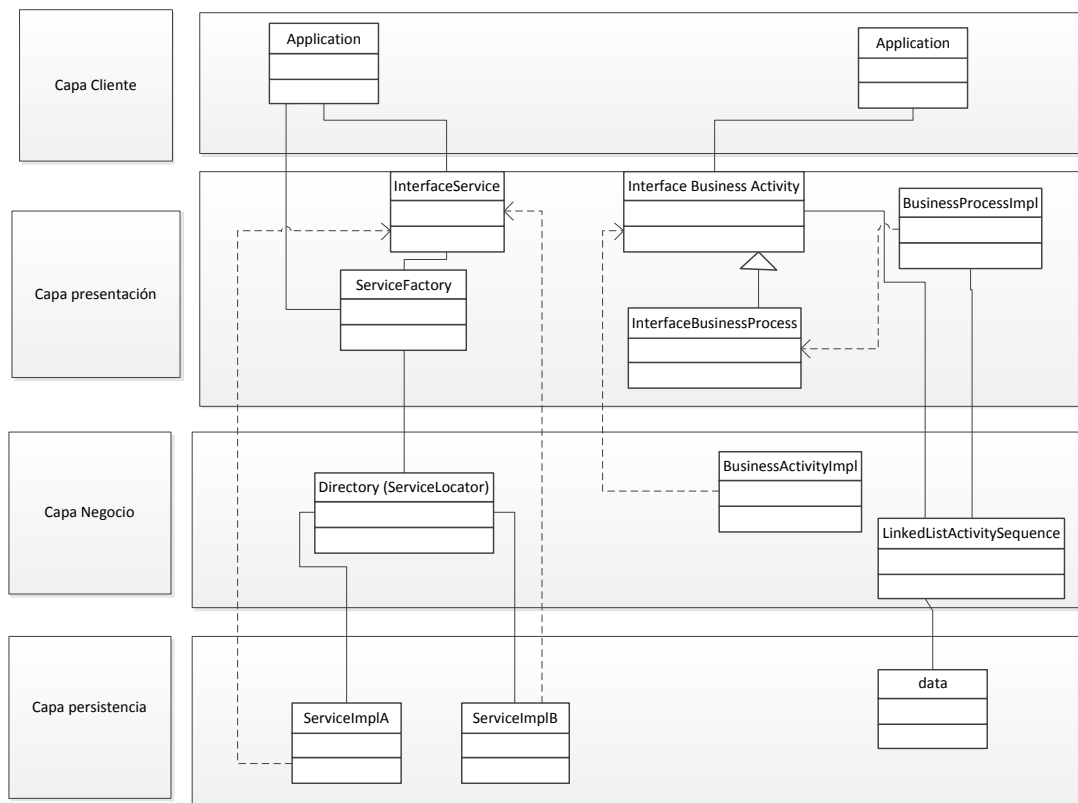


Figura 7. Diseño detallado (nivel 1): Patron n-tier mas patrones web service. Autor: Daniel Cáceres

La figura 7 nos permite observar del lado izquierdo se observa el nombre de cada una de las capas, en el centro se ve la estructura del patrón Service Factory y al lado derecho se aprecia la figura del patrón Business Process (composition).

Profundizando el diseño, relacionamos también más patrones dentro de cada uno de los de Web services ya definidos. De allí, que se relacionen de la siguiente forma:

- Para el service Factory
 - con la clase Application, el patrón Intercepting Filter
 - con la clase InterfaceService, el patrón Front Controller
 - con la clase ServiceFactory, el patrón Service to worker
 - con la clase Directory, el patrón Service Locator
 - con la clase ServiceImplA y ServiceImplB, el patrón Service Activator.
- Para el Business Process (composition)
 - con la clase Application, el patrón Intercepting Filter
 - con la clase InterfaceBusinessActivity, el patrón Front Controller
 - con la clase InterfaceBusinessProcess, el patrón Front Controller
 - con la clase BusinessProcessImpl, el patrón Dispatcher View

- con la clase BusinessActivityImpl, el patrón Business Delegate
- con la clase LinkedListActivitySequence, el patrón Value List Handler
- con la clase Data, el patrón Transfer Object y el Composite Entity

La descripción de cada patrón se presenta en la siguiente tabla:

CAPA DEL CLIENTE	
Decorating Filter / Intercepting Filter	Un objeto que está entre el cliente y los componentes Web. Este procesa las peticiones y las respuestas.
CAPA DE PRESENTACION	
Front Controller/ Front Component	Un objeto que acepta todos los requerimientos de un cliente y los direcciona a manejadores apropiados. El patrón Front Controller podría dividir la funcionalidad en 2 diferentes objetos: el Front Controller y el Dispatcher. En ese caso, El Front Controller acepta todos los requerimientos de un cliente y realiza la autenticación, y el Dispatcher direcciona los requerimientos a manejadores apropiada.
Service Worker To	Es como el patrón de diseño MVC con el Controlador actuando como Front Controller pero con una cosa importante: aquí el Dispatcher (el cual es parte del Front Controller) usa View Helpers a gran escala y ayuda en el manejo de la vista.
Dispatcher View	Es como el patrón de diseño MVC con el controlador actuando como Front Controller pero con un asunto importante: aquí el Dispatcher (el cual es parte del Front Controller) no usa View Helpers y realiza muy poco trabajo en el manejo de la vista. El manejo de la vista es manejado por los mismos componentes de la Vista.
CAPA DE NEGOCIO	
Business Delegate	Un objeto que reside en la capa de presentación y en beneficio de los otros componentes de la capa de presentación llama a métodos remotos en los objetos de la capa de negocios.
Value List Handler/ Page-by-Page Iterator/ Paged List	Es un objeto que maneja la ejecución de consultas SQL, caché y procesamiento del resultado. Usualmente implementado como beans de sesión.
Service Locator	Consiste en utilizar un objeto Service Locutor para abstraer toda la utilización JNDI y para ocultar las complejidades de la creación del contexto inicial, de búsqueda de objetos home EJB y recreación de objetos EJB. Varios clientes pueden reutilizar el objeto Service Locutor para reducir la complejidad del código, proporcionando un punto de control.

CAPA DE PERSISTENCIA	
Service Activator	Se utiliza para recibir peticiones y mensajes asíncronos de los clientes. Cuando se recibe un mensaje, el Service Activator localiza e invoca a los métodos de los componentes de negocio necesarios para cumplir la petición de forma asíncrona.
Value Object/ Data Transfer Object/ Replicate Object	Un objeto serializable para la transferencia de datos sobre la red.
Aggregate Entity composite entity /	Un bean entidad que es construido o es agregado a otros beans de entidad.

Tabla 12. Patrones JEE adecuados por capa. Fuente [18]

De esta manera, el diseño más detallado, a nivel 2, quedaría construido de la siguiente manera:

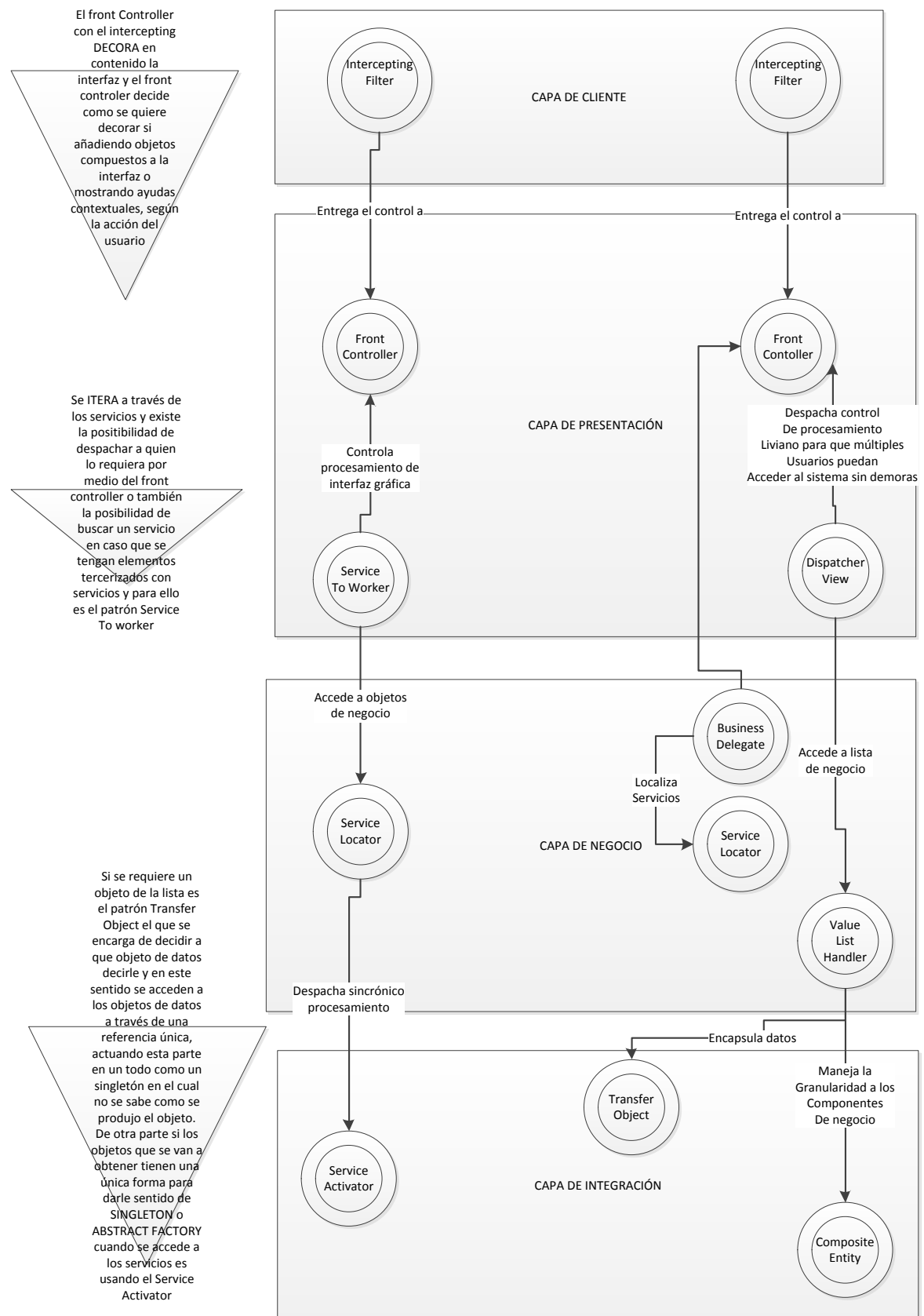


Figura 8. Diseño de detallado (nivel 2) del voto electrónico.

8. Diseño de la base de datos para el voto electrónico

El diseño de la base de datos se crea teniendo en cuenta los actores y elementos de interacción para el sistema de voto electrónico. A continuación se muestran los elementos que se interpretaron para la construcción del diseño de base de datos tanto para el proceso de inscripciones como para el proceso de votación, donde se bosquejó un estado ideal donde los partidos políticos estén interconectados con la registraduría para la asignación de avales de candidatos y el diagrama lógico de la base de datos planteada:

8.1. Diseño lógico de la base de datos para el proceso de inscripción de electores y candidatos

Entidad	Elector
Num_cedula (PK)	Integer
Nombres	Varchar(50)
Apellidos	Varchar(50)
Lugar_exp_ced	Varchar(50)
Direccion_residencia	Varchar(50)
Teléfono	Integer
Es_jurado	Boolean (1)

Tabla 13. Elector

Entidad	Cierre_Proceso_electoral
Fecha_cierre	Date
Hora_cierre	Time
Num_proc_elec (PF)	Integer

Tabla 14. Cierre proceso electoral

Entidad	Inicio_Proceso_electoral
Fecha_inicio	Date
Hora_inicio	Time
Num_proc_elec (PF)	Integer

Tabla 15. Inicio Proceso Electoral

Entidad	Proceso_electoral
Nombre_proc_elec	Varchar(50)
Num_proc_elec (PK)	Integer

Tabla 16. Proceso electoral

Entidad	Municipio
Cod_mpio (PK)	Integer
Nom_mpio	Varchar (50)
Cod_dpto (PF)	Integer
Num_hab	Integer

Tabla 17. Municipio

Entidad	Departamento
Cod_dpto (PK)	Integer
Nombre_dpto	Varchar (50)
Prom_votantes	NUMERIC (9,2)
Pot_elect	Integer
Region	Varchar(50)

Tabla 18. Departamento

Entidad	Inscripcion_elector
Num_form_insc_elector (PK)	Integer
Ultimo_lugar_votacion	Varchar (50)
Num_cedula (F)	Integer
Cod_mpio (PF)	Integer
Cod_dpto (PF)	Integer
Num_proc_elec (PF)	Integer
Num_proc_elec1 (PF)	Integer

Tabla 19. Inscripción elector

Entidad	Inscripcion_candidato
Num_form_insc_candidato (PK)	Integer
Certificado_residencia_notarial	Boolean (1)
Num_cedula (F)	Integer
Cod_mpio (PF)	Integer
Cod_dpto (PF)	Integer
Num_proc_elec (PF)	Integer
Num_proc_elec1 (PF)	Integer
Id_cargo (PF)	Integer
Ultimo_lugar_votacion	Varchar (50)
Num_aval (PF)	Integer

Tabla 20. Inscripción candidato

Entidad	Cargo_Candidato
Id_cargo	Integer
Nombre_cargo	Varchar (50)

Tabla 21. Cargo candidato

Entidad	Aval
Fecha_aval	Date
Hora_aval	Time
Num_tipo (FK)	Integer
Num_aval (PK)	Integer
Num_partido (FK)	Integer

Tabla 22. Aval

Entidad	Partido_politico
Num_partido	Integer
Nombre_partido	Varchar (50)

Tabla 23. Partido Político

Entidad	Tipo_Aval
Num_tipo	Integer
Nombre_aval	Varchar (50)

Tabla 24. Tipo Aval

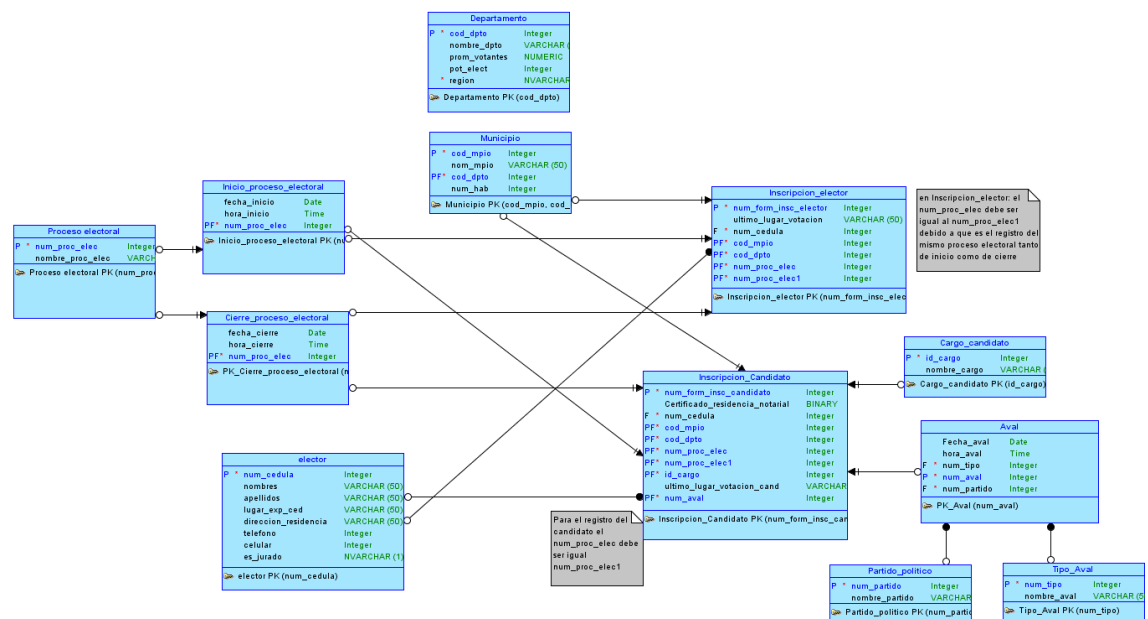


Figura 9. Modelo Lógico del proceso de inscripción de electores y candidatos

Para ver la figura completa ir al anexo

8.2. Diseño lógico de la base de datos para el proceso de votación

Debido a que el modelo base de datos para votación utiliza algunas de las tablas que se emplearon en el modelo de inscripción no se repetirán.

Entidad	Puesto
Cod_pto (PK)	Integer
Nombre_pto	NVarchar
Cod_mpio1 (PF)	Integer
Cod_dpto1 (PF)	Integer
Zona	Integer

Tabla 25. Puesto

Entidad	Mesa
Num_mesa (PK)	Integer
Pot_votantes	Integer
Cod_pto1 (PF)	Integer
Cod_mpio1 (PF)	Integer
Cod_dpto1 (PF)	Integer

Tabla 26. Mesa

Entidad	Boletín
Numero_boletin (PK)	Integer
Hora	Time
Fecha	Date
Num_mesa (PF)	Integer
Cod_pto1 (PF)	Integer
Cod_mpio1 (PF)	Integer
Cod_dpto1 (PF)	Integer
Mesa_informadas	Integer
Id_corporacion (F)	Integer

Tabla 27. Boletín

Entidad	Cargo_jurado
Id_cargo (PK)	Integer
Nombre_cargo	Varchar (50)

Tabla 28. Cargo jurado

Entidad	Jurado
Material	Varchar
Hora_entrada	Time
Hora_salida	time
Id_cargo (PF)	Integer
Fecha	Date

Tabla 29. Jurado

Entidad	Participantes
Cedula (PK)	Integer
Nombre	Varchar(50)
Num_participante	Integer

Tabla 30. Participantes

Entidad	Otro tipo
Id_otrotipo (PK)	Integer
Nombre_puesto	Varchar(50)

Tabla 31. Otro tipo

La entidad inscrip se crea como una copia de la entidad inscripción_elector solo que no con la totalidad de los atributos

Entidad	Inscrip
Num_mesa (PF)	Integer
Cod_pto1 (PF)	Integer
Cod_mpio1 (PF)	Integer
Cod_dpto1 (PF)	Integer
Num_cedula (F)	Integer
Num_form_insc_elector (PK)	Integer

Tabla 32. Inscrip

La entidad Inscripcion_candidato del modelo de votación es la misma entidad inscripcion_candidato del modelo de inscripción, solo que no se dibujó con sus respectivas conexiones para facilidad de entendimiento.

Entidad	Candidato
Num_candidato (PK)	Integer
Cod_partido (PF)	Integer
Num_form_insc_candidato (F)	Integer
Cod_mpio (F)	Integer
Cod_dpto (F)	Integer
Id_cargo (F)	Integer
Num_aval (F)	Integer

Tabla 33. Candidato

La entidad partido se duplica ya que en el momento de la jornada de votación pueden haberse realizado alianzas que no estaban cuando se realizó la inscripción pero que posteriormente se efectuaron bajo coordinación de la registraduría en su sede central (Bogotá)

Entidad	Partido
Cod_partido (PK)	Integer
Nombre partido	Varchar (50)

Tabla 34. Partido

Entidad	Candidato_consulta
Cod_candidato_con (PK)	Integer

Tabla 35. Candidato consulta

Entidad	Voto_alcaldia
Cod_partido (PF)	Integer
Num_candidato (PF)	Integer

Tabla 36. Voto Alcaldía

Entidad	Voto_Concejo
Cod_partido (PF)	Integer
Num_candidato (PF)	Integer

Tabla 37. Voto Concejo

Entidad	Voto_consulta
Cod_candidato_con (PF)	Integer

Tabla 38. Voto consulta

Entidad	Voto
Num_voto (PK)	Integer
Voto_blanco	Boolean (1)
Num_mesa (PF)	Integer
Cod_pto1 (PF)	Integer
Cod_mpio1 (PF)	Integer
Cod_dpto1 (PF)	Integer
Id_corporacion (PF)	Integer
Fecha	Date
hora	Time

Tabla 39. Voto

La tabla Corporación se crea para poder facilitar la consulta de boletines. El término de corporación se da para los cargos a escoger que hallan: presidencia, referendos, consultas partidistas, alcaldía, concejo, gobernación, asamblea, cámara y senado.

Entidad	Corporacion
Id_corporacion (PK)	Integer
Tipo_corporacion	Varchar (50)

Tabla 40. Corporación

[illegible]

Figura 10. Modelo lógico del proceso de votación

9. Validación de la arquitectura de software

9.1. Validación según ATAM

El método Architecture Tradeoff Analysis Method – ATAM - según Paul Clements et. Al [19][20] es un método que evalúa las consecuencias de las decisiones arquitectónicas a la luz de los requisitos de los atributos de calidad. ATAM trabaja sobre tres áreas: la noción de estilos arquitectónicos, las comunidades de análisis de atributos de calidad, y el método de análisis de arquitecturas de software (Software Architecture Analysis Method – SACAM).

Una de las implicaciones de usar ATAM es que puede ser aplicado al principio del ciclo de desarrollo de software. De allí que las mayores metas de ATAM sean:

- Obtener y refinar una declaración precisa del manejo de los requisitos de los atributos de calidad de la arquitectura.
- Obtener y refinar una declaración precisa de las decisiones de diseño arquitectónico.

- c) Evaluar las decisiones de diseño arquitectónico para determinar si dan una respuesta satisfactoria a las exigencias de calidad.

9.2. Presentación de la arquitectura

Los pasos principales del método ATAM son:

Paso	Actividad	Partes implicadas
1	Presentación de ATAM	Equipo de Evaluación Equipo de desarrollo Representación cualificada del cliente
2	Presentación de los factores de negocio	Equipo de Evaluación Equipo de desarrollo Representación cualificada del cliente
3	Presentación de la arquitectura	Equipo de Evaluación Equipo de desarrollo Representación cualificada del cliente
4	Identificación de los estilos arquitectónicos	Equipo de Evaluación Equipo de desarrollo Representación cualificada del cliente
5	Generación del árbol de utilidad	Equipo de Evaluación Equipo de desarrollo Representación cualificada del cliente
6	Análisis de los estilos arquitectónicos	Equipo de Evaluación Equipo de desarrollo Representación cualificada del cliente
7	Tormenta de ideas y priorización de escenarios	Todas
8	Análisis de los estilos arquitectónicos	Equipo de Evaluación Equipo de desarrollo Representación cualificada del cliente
9	Presentación de resultados	Todas

Tabla 41. Pasos principales del método ATAM. Autor: Daniel Cáceres

9.3. Presentación de los factores de negocio

Identificando los requerimientos iniciales del sistema se puede visualizar la relación entre estos requerimientos, los atributos de calidad del sistema y los elementos que están relacionados con la consecución de estos atributos a nivel de la arquitectura. Este trabajo dará luces para establecer los escenarios que se tendrán para cada factor de negocio. Lo importante del método es que una vez se tenga formalizado que el diseño global de la arquitectura ha sido completamente válido a la luz de los requerimientos, se procede a elaborar los elementos de diseño para verificar que siguen cumpliendo las especificaciones

originales y para este paso se puede aplicar nuevamente y recursivamente el método. Sin embargo, este tipo de situaciones generalmente se acostumbran a dejar cuando se está haciendo implementaciones modulares y se tienen que modificar diseños concretos específicos del lenguaje de programación en el que se implemente.

Debido a la extensión de los requerimientos del sistema, el análisis se realizara sobre los casos de uso que reúnen muchos de los requerimientos del sistema.

Casos de uso	Atributo de calidad asociado al requerimiento inicial. El que más prima	Proceso de negocio que se relaciona con dicho requerimiento	Elementos de arquitectura que intervienen en el sistema.	Escenarios más relevantes al interior
Almacenar formularios en DB	Modularidad	Aplicación Registro candidato	Service Factory Pattern	Esc1: Al momento de ingresar el formulario otro usuario desea consultarlo y modificarlo al mismo tiempo Esc2: En caso de migrarse la DB a la que acceden los servicios
Almacenar registro candidatos en DB	Modularidad	Aplicación Registro candidato	Service Factory Pattern	Esc1: Al momento de ingresar el registro, otro usuario desea consultarlo y modificarlo al mismo tiempo Esc2: Al momento de ingresar el registro, otro usuario desea consultarlo y modificarlo al mismo tiempo
Almacenar registro electores en DB	Modularidad	Aplicación Registro elector	Service Factory Pattern	Esc1: La creación de un registro de elector que ya

				existe
Almacenar reportes	Capacidad de expansión	Aplicación resultado escrutinio	Business Process	Esc1: Antes de haber cerrado la jornada de inscripciones de candidatos o de electores, se requiere tener un informe sobre los registros efectuados
Almacenar conteo preliminar	Capacidad de expansión	Aplicación resultado escrutinio	Business Process	Esc1: Antes de haber cerrado la jornada de votaciones se requiere tener un informe sobre los votos hasta ahora escrutados
Almacenar votos del sistema	Simplicidad	Aplicación votación	Business Process	Esc1: Se da inicio a la jornada de votación y se presenta un elector que no está inscrito en el municipio
Bajar Servicio Jornada de Votación	Tolerancia al error	Aplicación votación	Business Process	Esc1: Se presentan altercados de orden público en el puesto de votación y se requiere bajar los servicios antes de la hora estipulada
Bajar Servicio Post-Elección	Tolerancia al error	Aplicación resultado escrutinio	Business Process	Esc1: Alternados de orden público mientras se hace el conteo de los votos
Bajar servicio de validación biométrica y código de barras	Tolerancia al error	Aplicación votación	Business Process	Esc1: El sistema SAGEM tiene fallos o hay reportes de error del sistema SAGEM

Cerrar Jornada Post-Elección	Operatividad	Aplicación resultado escrutinio	Business Process	Esc1: El servicio jornada de votación fue abierto después de su cierre
Cerrar Jornada de Votación	Operatividad	Aplicación votación	Business Process	Esc1: Intento de ingreso de datos a la aplicación después de cerrada la jornada
Cerrar Proceso De Inscripciones candidatos	Operatividad	Aplicación Registro candidato	Service Factory Pattern	Esc1: Intento de ingreso del nuevo registro de candidato después de cerrar el proceso
CerrarProceso De Inscripciones electores	Operatividad	Aplicación Registro elector	Service Factory Pattern	Esc1: Intento de ingreso del nuevo registro de elector después de cerrar el proceso
Crear DB de reportes	Modularidad	Aplicación resultado escrutinio	Business Process	Esc1: Se necesita una nueva base de datos para una nueva jornada electoral.
Crear DB de votos	Modularidad	Aplicación votación	Business Process	Esc1: Se necesita una nueva base de datos para una nueva jornada electoral
Crear formulario E-24/E-26	Usabilidad	Aplicación resultado escrutinio	Business Process	Es1: Generar un formulario de un proceso no estipulado
Crear una DB de registro electores	Modularidad	Aplicación Registro elector	Service Factory Pattern	Esc1: Intento de copia de registro de elector
Crear una DB de registro candidatos	Modularidad	Aplicación Registro candidato	Service Factory Pattern	Esc1: Un candidato se va a registrar como elector en otro municipio

Generar certificado electoral	Expendability	Aplicación resultado escrutinio	Business Process	Esc1: Un usuario que se inscribió pero no votó reclama el certificado electoral
Generar certificado inscripción alcaldía	Expendability	Aplicación resultado escrutinio	Business Process	Esc1: Un elector que se inscribió reclama el certificado de inscripción alcalde
Generar certificado inscripción concejo	Expendability	Aplicación resultado escrutinio	Business Process	Esc1: Un elector que se inscribió reclama el certificado de inscripción concejo
Generar certificado inscripción electoral	Expendability	Aplicación resultado escrutinio	Business Process	Esc1: Un elector que no se inscribió reclama el certificado de inscripción electoral
Habilitar acceso a la interface de votación	Tolerancia al error	Aplicación votación	Business Process	Esc1: Un usuario no registrado intenta acceder a la interface de votación de forma remota
Iniciar proceso Inscripciones candidatos	Usabilidad	Aplicación Registro candidato	Service Factory Pattern	Esc1: Se realiza la inscripción de un candidato, pero la registraduría que la inscripción de electores aún no había iniciado
Iniciar Proceso Inscripciones Electores	Usabilidad	Aplicación Registro elector	Service Factory Pattern	Esc1: Un candidato desea ser registrado con sus datos de elector
Iniciar Jornada de Votación	Tolerancia al error	Aplicación votación	Business Process	Esc1: Se intenta iniciar la jornada de votación sin haber cerrado el

				proceso de registro de candidatos y registro de electores
Iniciar Servicio Jornada Votación	Tolerancia al error	Aplicación votación	Business Process	Esc1: Se intenta acceder al servicio web de jornada de votación sin haber iniciado el proceso jornada votación
Iniciar Jornada Post-Elección	Tolerancia al error	Aplicación resultado escrutinio	Business Process	Esc1: Se intenta iniciar la jornada post-elección sin haber cerrado la jornada de votación
Iniciar Servicio Jornada Post-Elección	Tolerancia al error	Aplicación resultado escrutinio	Business Process	Esc1: Se intenta acceder al servicio web de jornada post-elección sin haber iniciado el proceso jornada post-elección
Iniciar servicio de validación biométrica y de código de barras	Tolerancia al error	Aplicación votación	Business Process	Esc1: Se intenta registrar un elector sin validar su identidad
Monitorear el sistema de votación	Tolerancia al error	Aplicación votación	Business Process	Esc1: Hay un puesto de votación que está fallando.
Validar el elector en jornada de votación	Generalidad	Aplicación Registro elector	Service Factory Pattern	Esc1: Se presenta un elector con una cedula falsa y pretende votar
Validar mediante lector biométrico y de código de barras en jornada de inscripción	Formación	Aplicación votación	Service Factory Pattern	Esc1: Se presenta un elector con una cedula falsa y pretende inscribirse.

Tabla 42. Aplicación de ATAM sobre la arquitectura de sistema del voto electrónico

9.4. Análisis de resultados

Con base en la tabla anterior, se procede a dar explicación de cómo la arquitectura y los atributos de calidad definidos si son importantes para la mayoría de escenarios propuestos:

Patrón	Service Factory Pattern	
Caso de uso	Almacenar formularios en DB	
Lo que permite el patrón	Permite utilizar el patrón mediador para mitigar las diferencias entre servicios	Cuando el tiempo para una transición llega usted podría configurar la factoría para localizar el nuevo sistema de persistencia de los datos
Situaciones Excepcionales	Al momento de ingresar el formulario otro usuario desea consultarlo y modificarlo al mismo tiempo	En caso de migrarse la DB a la que acceden los servicios
Justificación	Las diferencias en los servicios de consulta y de modificación pueden ser mitigadas mediante el patrón mediator	Lo formularios acceden a una factoría que permite localizar el nuevo sistema de persistencia de datos
Resultado	Tanto las consultas como las modificaciones pueden ser realizadas desde dos usuarios diferentes	El sistema se libera de tener dependencia con respecto a la localización de la base de datos y esto permite agilidad en futuras migraciones de los sistemas de persistencia

Tabla 43. Patrón Service Factory versus CU Almacenar formularios en DB

Patrón	Service Factory Pattern	
Caso de uso	Almacenar Registro candidatos	
Lo que permite	Permite utilizar el patrón mediador para mitigar las diferencias entre servicios	Implica que se deben mantener estándares para permitir el desarrollo de interfaces propias como para los clientes
Situaciones Excepcionales	Al momento de ingresar el registro, otro usuario desea consultarlo y modificarlo al mismo tiempo	Al momento de ingresar el registro, otro usuario desea consultarlo y modificarlo al mismo tiempo
Justificación	Las diferencias en acceso y modificación pueden ser mitigadas mediante el patrón mediator	Mediante interfaces se puede independizar el acceso a la consulta del registro y a la modificación
Resultado	Tanto el acceso como las modificaciones pueden ser realizadas desde dos	No restringe el rendimiento de un servicio cuando acceda ya sea para consulta o modificación

	usuarios diferentes
--	---------------------

Tabla 44. Patrón Service Factory versus CU Almacenar Registro Candidatos

Patrón	Service Factory Pattern
Caso de uso	Almacenar Registro Electores
Lo que permite	Se podría usar para relacionar objetos de negocio con colecciones de objetos
Situaciones Excepcionales	La creación de un registro de elector que ya existe
Justificación	El Service Factory permite establecer relaciones directas con la colección de objetos creados previamente evitando crear redundancias en la capa de persistencia
Resultado	La aplicación del Service Factory permite bajos niveles de redundancia en los registros de la base de datos

Tabla 45. Patrón Service Factory versus CU Almacenar Registro Electores

Patrón	Business Process (Composition) Pattern
Caso de uso	Almacenar Conteo Preliminar
Lo que permite	Tiene un tipo de modelo proxy para conectar las representaciones de datos a través de una red y para acceder a los datos el cliente del web service debe recrear el modelo de objeto en el servidor en su propio lenguaje
Situaciones Excepcionales	Antes de haber cerrado la jornada de votaciones se requiere tener un informe sobre los votos hasta ahora escrutados
Justificación	Los clientes haciendo uso de las capacidades del patrón pueden generar copias del modelo de objetos del servidor
Resultado	Se pueden obtener copias temporales aprovechando el uso del patrón

Tabla 46. Patrón Business Process (Composition) versus CU Almacenar Conteo Preliminar

Patrón	Business Process (Composition) Pattern
Caso de uso	Almacenar Reportes
Lo que permite	Tiene un tipo de modelo proxy para conectar las representaciones de datos a través de una red y para acceder a los datos el cliente del web service debe recrear el modelo de objeto en el servidor en su propio lenguaje
Situaciones Excepcionales	Antes de haber cerrado la jornada de inscripciones de candidatos o de electores, se requiere tener un informe sobre los registros efectuados
Justificación	Los clientes haciendo uso de las capacidades del patrón pueden generar copias del modelo de objetos del servidor que contengan información sobre los registros efectuados

Resultado	Se pueden obtener reportes temporales aprovechando el uso del patrón
------------------	--

Tabla 47. Patrón Business Process (Composition) versus CU Almacenar Reportes

Patrón	Business Process (Composition) Pattern
Caso de uso	Almacenar votos del sistema
Lo que permite	Hay pasos o entradas en los proceso de negocio que involucran la coordinación de procesos y actividades previas
Situaciones Excepcionales	Se da inicio a la jornada de votación y se presenta un elector que no está inscrito en el municipio
Justificación	Para poder ejecutar el almacenamiento del voto de cualquier elector antes que este acceda al sistema de votación se debe verificar en la base de datos de inscripción de electores que el elector se halla inscrito en el municipio
Resultado	El sistema almacena el voto siempre y cuando el elector haya sido validado

Tabla 48. Patrón Business Process (Composition) versus CU Almacenar Votos del sistema

Patrón	Business Process (Composition) Pattern
Caso de uso	Bajar Servicio Jornada de Votación
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web. / La intención del patrón es ilustrar y proveer una guía para combinar las actividades de negocio en un único y consumible servicio web con una interface bien definida
Situaciones Excepcionales	Se presentan altercados de orden público en el puesto de votación y se requiere bajar los servicios antes de la hora estipulada
Justificación	Las actividades y procesos que componen la aplicación de votación deben ser tenidas en cuenta para bajar el servicio de la jornada de votación
Resultado	Los servicios que se encuentren en ejecución se cierran adecuadamente y posteriormente el sistema cierra la aplicación de votación

Tabla 49. Patrón Business Process (Composition) versus CU Bajar servicio jornada de votación

Patrón	Business Process (Composition) Pattern
Caso de uso	Bajar Servicio Post-Elección
Lo que permite	La intención del patrón es ilustrar y proveer una guía para combinar las actividades de negocio en un único y consumible servicio web con una interface bien definida
Situaciones Excepcionales	Alternados de orden público mientras se hace el conteo de los votos
Justificación	Dado que se puede acceder mediante un único servicio web no importa que el lugar de acceso al servicio no esté físicamente donde se presentan los problemas de orden público y donde se encuentra la base de datos donde se van a almacenar los datos

Resultado	Se puede realizar el conteo de los votos desde otro lugar
------------------	---

Tabla 50. Patrón Business Process (Composition) versus CU Bajar servicio Post-Elección

Patrón	Business Process (Composition) Pattern
Caso de uso	Bajar servicio de validación biométrica y código de barras
Lo que permite	Hay pasos o entradas en los proceso de negocio que involucran la coordinación de procesos y actividades externas
Situaciones Excepcionales	El sistema SAGEM tiene fallos o hay reportes de error del sistema SAGEM
Justificación	Si no se está usando la validación con el sistema SAGEM mantener conexiones con dicho sistema baja el rendimiento
Resultado	Mejoramiento del rendimiento

Tabla 51. Patrón Business Process (Composition) versus CU Bajar servicio de validación biométrica y código de barras

Patrón	Business Process (Composition) Pattern
Caso de uso	Cerrar Jornada Post-Elección
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web
Situaciones Excepcionales	El servicio jornada de votación fue abierto después de su cierre
Justificación	Para poderse cerrar la jornada post-elección tuvo que haberse ejecutado el servicio jornada post-elección que depende directamente de haber cerrado el servicio jornada de votación
Resultado	El sistema anularía cualquier modificación realizada sobre la base de datos después de la primera fecha de cierre del servicio jornada de votación

Tabla 52. Patrón Business Process (Composition) versus CU Cerrar jornada Post-Elección

Patrón	Business Process (Composition) Pattern
Caso de uso	Cerrar Jornada de Votación
Lo que permite	Hay pasos o entradas en los proceso de negocio que involucran la coordinación de procesos y actividades externas
Situaciones Excepcionales	Intento de ingreso de datos a la aplicación después de cerrada la jornada
Justificación	El hecho de realizar el cierre de la jornada evita fraudes en la jornada electoral ya que esta acción de cierre de la jornada funcionara como impedimento de seguridad ante los próximos eventos de ingreso de datos después de cerrada la jornada
Resultado	El sistema no permite ingresos de nuevos registros a la base de datos una vez cerradas las jornadas electorales

Tabla 53. Patrón Business Process (Composition) versus CU Cerrar jornada de votación

Patrón	Service Factory Pattern
Caso de uso	Cerrar Proceso De Inscripciones candidatos
Lo que permite	Aislar la selección de servicios y la lógica de instanciación
Situaciones Excepcionales	Intento de ingreso del nuevo registro de candidato después de cerrar el proceso
Justificación	Usando el patrón Service Factory se facilita la restricción de acceso al servicio web, en este caso sería el servicio de registro de candidato, por lo tanto no habría forma de ingresar nuevos registros de candidatos
Resultado	El sistema no permite después del cierre del proceso el almacenamiento de nuevas inscripciones o modificaciones del candidato

Tabla 54. Patrón Service Factory versus CU Cerrar proceso de inscripciones candidatos

Patrón	Service Factory Pattern
Caso de uso	Cerrar Proceso De Inscripciones electores
Lo que permite	Aislar la selección de servicios y la lógica de instanciación
Situaciones Excepcionales	Intento de ingreso del nuevo registro de elector después de cerrar el proceso
Justificación	Usando el patrón Service Factory se facilita la restricción de acceso al servicio web, en este caso sería el servicio de registro de elector, por lo tanto no habría forma de ingresar nuevos registros de electores
Resultado	El sistema no permite después del cierre del proceso el almacenamiento de nuevas inscripciones de elector

Tabla 55. Patrón Service Factory versus CU Cerrar proceso de inscripciones electores

Patrón	Business Process (Composition) Pattern
Caso de uso	Crear DB de reportes
Lo que permite	Una faceta del proceso de negocio direcciona la estructura de un proceso de negocio y la otra faceta direcciona como estandarizar las interfaces para los proceso de negocio
Situaciones Excepcionales	Se necesita una nueva base de datos para una nueva jornada electoral
Justificación	El hecho de crear una nueva base de datos no afecta la estructura lógica del proceso de negocio
Resultado	La BD creada es fácilmente direccionable por los servicios gracias al manejo del patrón business process composition

Tabla 56. Patrón Business Process (Composition) versus CU Crear DB de reportes

Patrón	Business Process (Composition) Pattern
Caso de uso	Crear DB de votos

Lo que permite	Una faceta del proceso de negocio direcciona la estructura de un proceso de negocio y la otra faceta direcciona como estandarizar las interfaces para los proceso de negocio
Situaciones Excepcionales	Se necesita una nueva base de datos para una nueva jornada electoral
Justificación	El hecho de crear una nueva base de datos no afecta la estructura lógica del proceso de negocio
Resultado	La BD creada es fácilmente direccionable por los servicios gracias al manejo del patrón business process composition

Tabla 57. Patrón Business Process (Composition) versus CU Crear DB de votos

Patrón	Business Process (Composition) Pattern
Caso de uso	Crear formulario E-24/E-26
Lo que permite	El proceso de negocio es una abstracción de la lógica compleja de negocio que consiste de uno o más actividades de negocio. La lógica y el flujo de las actividades y el flujo que los datos toman a través de proceso de negocio
Situaciones Excepcionales	Generar un formulario de un proceso no estipulado
Justificación	El patrón permite ajustar dentro del flujo de los procesos de negocio que actividad o subproceso puede ser automatizado como es el caso de un formulario
Resultado	El sistema no permite la generación de formularios que no estén definidos y vinculados dentro del flujo de los procesos de negocio

Tabla 58. Patrón Business Process (Composition) versus CU Crear formulario E-24/E-26

Patrón	Service Factory Pattern
Caso de uso	Crear una DB de registro candidatos
Lo que permite	Cuando el tiempo para una transición llega usted podría configurar la factoría para localizar el nuevo sistema de persistencia de los datos
Situaciones Excepcionales	Un candidato se va a registrar como elector en otro municipio
Justificación	El patrón permite hacer la replicación de datos gracias a a configuración de una factoría para el registro desde la aplicación registro candidato a la aplicación registro elector
Resultado	Cuando se cree la BD de candidato se permita replicar parte de la información de este registro a la base de datos de electores. Por tanto el candidato que desee registrarse como elector en otro municipio no lo podrá hacer

Tabla 59. Patrón Service Factory versus CU Crear una DB de registro candidatos

Patrón	Service Factory Pattern
Caso de uso	Crear una DB de registro electores
Lo que permite	Aislar la selección de servicios y la lógica de instanciación

Situaciones Excepcionales	Intento de copia de registro de elector
Justificación	Este patrón permite que solo la información que se usa en el servicio pueda ser empleada para crear el registro del elector, evitando que esta información pueda ser duplicada o solicitada en otro servicio externo como si lo hace el registro candidato
Resultado	Solo mediante el servicio de registrar elector se puede crear dicho registro

Tabla 60. Patrón Service Factory versus CU Crear una DB de registro electores

Patrón	Business Process (Composition) Pattern
Caso de uso	Generar certificado electoral
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web
Situaciones Excepcionales	Un usuario que se inscribió pero no votó reclame el certificado electoral
Justificación	Gracias al patrón se puede establecer que el servicio de generar certificado sea una composición del servicio registrar elector y el servicio votación
Resultado	El sistema no permite la generación de certificados electorales a personas que no hayan sido registrados en el municipio para votar y que no hayan realizado la acción de votar

Tabla 61. Patrón Business Process (Composition) versus CU Generar certificado electoral

Patrón	Business Process (Composition) Pattern
Caso de uso	Generar certificado inscripción alcaldía
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web
Situaciones Excepcionales	Un elector que se inscribió reclama el certificado de inscripción alcalde
Justificación	Gracias al patrón se puede establecer que el servicio de generar certificado sea una composición del servicio registrar candidato
Resultado	El sistema no permite la generación de certificados de inscripción de candidato a la alcaldía a personas que no hayan sido registrados como candidatos

Tabla 62. Patrón Business Process (Composition) versus CU Generar certificado inscripción alcaldía

Patrón	Business Process (Composition) Pattern
Caso de uso	Generar certificado inscripción concejo
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web
Situaciones Excepcionales	Un elector que se inscribió reclama el certificado de inscripción concejo

Justificación	Gracias al patrón se puede establecer que el servicio de generar certificado sea una composición del servicio registrar candidato
Resultado	El sistema no permite la generación de certificados de inscripción de candidato al concejo a personas que no hayan sido registrados como candidatos

Tabla 63. Patrón Business Process (Composition) versus CU Generar certificado inscripción concejo

Patrón	Business Process (Composition) Pattern
Caso de uso	Generar certificado inscripción electoral
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web
Situaciones Excepcionales	Un elector que no se inscribió reclama el certificado de inscripción electoral
Justificación	Gracias al patrón se puede establecer que el servicio de generar certificado sea una composición del servicio registrar elector
Resultado	El sistema no permite la generación de certificados de inscripción electoral a personas que no hayan sido registrados como electores

Tabla 64. Patrón Business Process (Composition) versus CU Generar certificado inscripción electoral

Patrón	Business Process (Composition) Pattern
Caso de uso	Habilitar acceso a la interface de votación
Lo que permite	Una faceta del proceso de negocio direcciona la estructura de un proceso de negocio y la otra faceta direcciona como estandarizar las interfaces para los proceso de negocio
Situaciones Excepcionales	Un usuario no registrado intenta acceder a la interface de votación de forma remota
Justificación	El acceso de un usuario no registrado no está direccionado dentro de la estructura del proceso de negocio, ni mucho menos estandarizado para usar las interfaces del proceso
Resultado	El sistema no permite el acceso del usuario no registrado

Tabla 65. Patrón Business Process (Composition) versus CU Habilitar acceso a la interface de votación

Patrón	Service Factory Pattern
Caso de uso	Iniciar proceso Inscripciones candidatos
Lo que permite	Permite utilizar el patrón mediador para mitigar las diferencias entres servicios
Situaciones Excepcionales	Se realiza la inscripción de un candidato, pero la registraduría que la inscripción de electores aún no había iniciado
Justificación	El patrón service factory da flexibilidad en la integración del patrón mediator
Resultado	Dado el servicio de inscripción de candidato se pueda utilizar información parcial de su registro para usarla en el registro de elector mediante el patrón mediator

Tabla 66. Patrón Service Factory versus CU Iniciar proceso inscripciones candidatos

Patrón	Service Factory Pattern
Caso de uso	Iniciar Proceso Inscripciones Electores
Lo que permite	Aislar la selección de servicios y la lógica de instanciación
Situaciones Excepcionales	Un candidato desea ser registrado con sus datos de elector
Justificación	El hecho que un elector se registre no implica que tenga que hacer un registro como candidato, por lo tanto el registro para electores estará definido dentro de la temporalidad de duración del proceso
Resultado	El servicio web que realiza el registro del elector crea solamente en la base de datos de electores este registro

Tabla 67. Patrón Service Factory versus CU Iniciar proceso inscripciones electores

Patrón	Business Process (Composition) Pattern
Caso de uso	Iniciar Jornada de Votación
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web
Situaciones Excepcionales	Se intenta iniciar la jornada de votación sin haber cerrado el proceso de registro de candidatos y registro de electores
Justificación	La jornada de iniciación de jornada de votación no se puede realizar si no se ha llevado a cabo el proceso de registro de candidatos y de electores
Resultado	El sistema no debe permitir el inicio de la jornada de votación

Tabla 68. Patrón Business Process (Composition) versus CU Iniciar jornada de votación

Patrón	Business Process (Composition) Pattern
Caso de uso	Iniciar Servicio Jornada Votación
Lo que permite	Una faceta del proceso de negocio direcciona la estructura de un proceso de negocio y la otra faceta direcciona como estandarizar las interfaces para los proceso de negocio
Situaciones Excepcionales	Se intenta acceder al servicio web de jornada de votación sin haber iniciado el proceso jornada votación
Justificación	Debido a que no se ha iniciado el proceso de negocio de jornada votación (elemento esencial de la estructura de proceso) no se puede dar cabida a invocar el servicio web que permite iniciar la jornada de votación
Resultado	No se permite invocar el servicio web

Tabla 69. Patrón Business Process (Composition) versus CU Iniciar servicio jornada de votacion

Patrón	Business Process (Composition) Pattern
Caso de uso	Iniciar Jornada Post-Elección

Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web
Situaciones Excepcionales	Se intenta iniciar la jornada post-elección sin haber cerrado la jornada de votación
Justificación	Hasta que no se ha cerrado la jornada de votación no se puede iniciar la jornada post-elección
Resultado	El sistema no debe permitir el inicio de la jornada post-elección

Tabla 70. Patrón Business Process (Composition) versus CU Iniciar jornada Post-Elección

Patrón	Business Process (Composition) Pattern
Caso de uso	Iniciar Servicio Jornada Post-Elección
Lo que permite	Una faceta del proceso de negocio direcciona la estructura de un proceso de negocio y la otra faceta direcciona como estandarizar las interfaces para los proceso de negocio
Situaciones Excepcionales	Se intenta acceder al servicio web de jornada post-elección sin haber iniciado el proceso jornada post-elección
Justificación	Debido a que no se ha iniciado el proceso de negocio de jornada post-elección (elemento esencial de la estructura de proceso) no se puede dar cabida a invocar el servicio web que permite iniciar la jornada post-elección
Resultado	No se permite invocar el servicio web

Tabla 71. Patrón Business Process (Composition) versus CU Iniciar servicio jornada Post-Elección

Patrón	Business Process (Composition) Pattern
Caso de uso	Iniciar servicio de validación biométrica y de código de barras
Lo que permite	Tiene un tipo de modelo proxy para conectar las representaciones de datos a través de una red y para acceder a los datos el cliente del web service debe recrear el modelo de objeto en el servidor en su propio lenguaje
Situaciones Excepcionales	Se intenta registrar un elector sin validar su identidad
Justificación	Dado que el patrón permite establecer la conexión con el servidor SAGEM para la aplicación morphocheck mediante un modelo proxy, si no establece la conexión el sistema no deberá permitir el registro de ningún elector
Resultado	No se permite el registro del elector

Tabla 72. Patrón Business Process (Composition) versus CU Iniciar servicio de validación biométrica y de código de barras

Patrón	Business Process (Composition) Pattern
Caso de uso	Monitorear el sistema de votación
Lo que permite	Un proceso de negocio en los servicios web es una composición de las actividades de negocio que pueden o no pueden ser otros servicios web

Situaciones Excepcionales	Hay un puesto de votación que está fallando
Justificación	El patrón business process composition permite el uso a nivel de diseño de patrones como el observer que son útiles para el monitoreo de los objetos de negocio
Resultado	La acción de monitorear refleja los puntos que tienen fallos en el sistema

Tabla 73. Patrón Business Process (Composition) versus CU Monitorear el sistema de votación

Patrón	Service Factory Pattern
Caso de uso	Validar el elector en jornada de votación
Lo que permite	Se podría usar para relacionar objetos de negocio con colecciones de objetos
Situaciones Excepcionales	Se presenta un elector con una cedula falsa y pretende votar
Justificación	La identificación del usuario se maneja como un objeto de negocio el cual se compara con la colección de objetos de negocio almacenados en el servidor SAGEM de la aplicación morphocheck
Resultado	El sistema para permitir la votación debe validar la identificación del elector

Tabla 74. Patrón Service Factory versus CU Validar el elector en jornada de votación

Patrón	Service Factory Pattern
Caso de uso	Validar mediante lector biométrico y de código de barras en jornada de inscripción
Lo que permite	Se podría usar para relacionar objetos de negocio con colecciones de objetos
Situaciones Excepcionales	Se presenta un elector con una cedula falsa y pretende inscribirse
Justificación	La identificación del usuario se maneja como un objeto de negocio el cual se compara con la colección de objetos de negocio almacenados en el servidor SAGEM de la aplicación morphocheck
Resultado	El sistema para registrar al elector no permite la creación del registro si la validación es falsa

Tabla 75. Patrón Service Factory versus CU Validar mediante lector biométrico y de código de barras en jornada de inscripción

9.5. Prototipo de interfaces de usuario

Debido a que según la ciencia basada en el diseño la solución que se cree para el problema real debe representarse como un artefacto, se implementó lo descrito anteriormente tanto de la arquitectura empresarial como de la arquitectura de software en un prototipo de interfaz de usuario donde se ajustaron los modelos de las vistas de Archimate a modelos en BPMN y se implementó en una herramienta llamada Bizagi para mantener el control de los procesos: tanto de inscripción de electores y candidatos como en el proceso de

votación. Todo esto con la funcionalidad de los procesos llevada a cabo con Web Services sobre el modelo de bases de datos.

9.5.1. Modelos BPMN

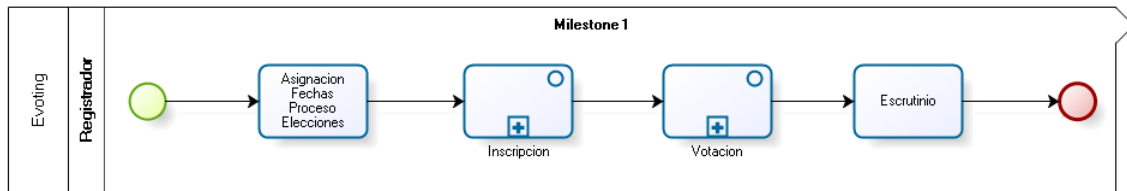


Figura 11. Modelo BPMN de la línea estratégica de votación

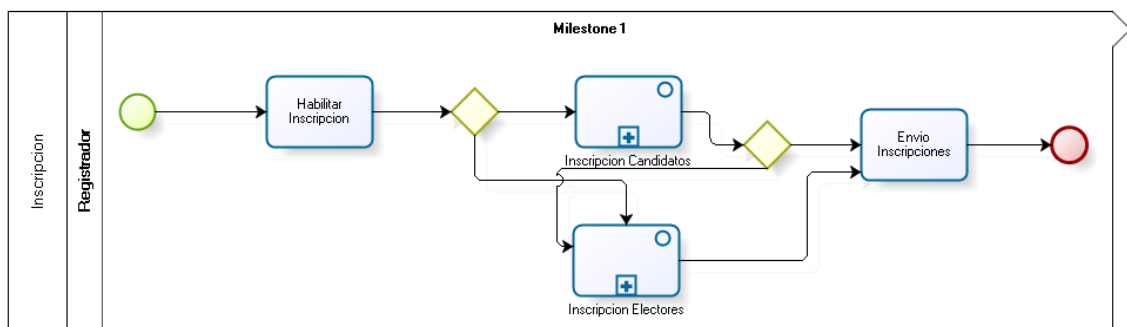


Figura 12. Modelo BPMN del subproceso de inscripción

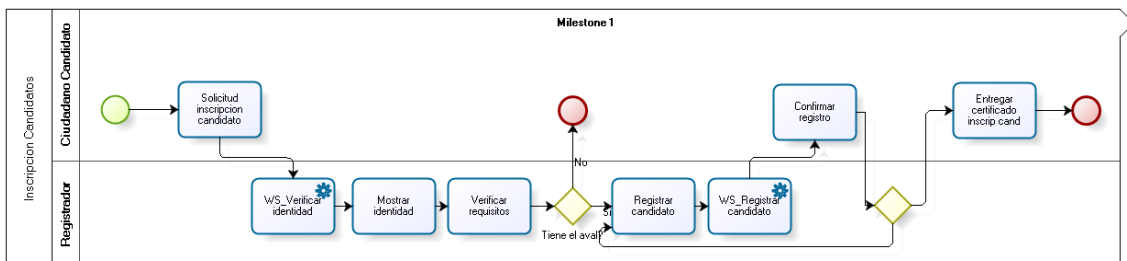


Figura 13. Modelo BPMN del subproceso de inscripción de candidatos

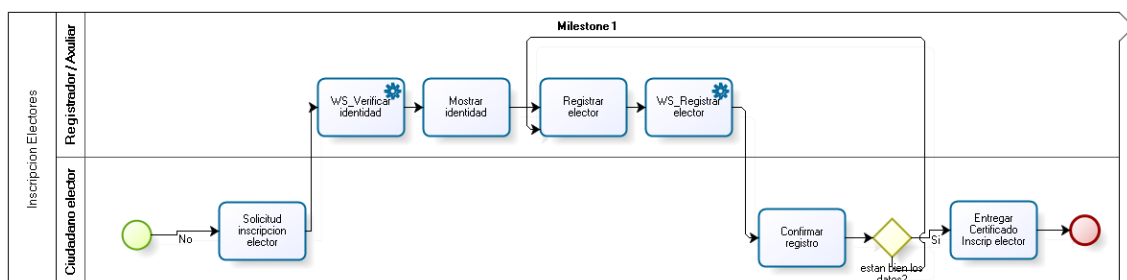


Figura 14. Modelo BPMN del subproceso de inscripción de electores

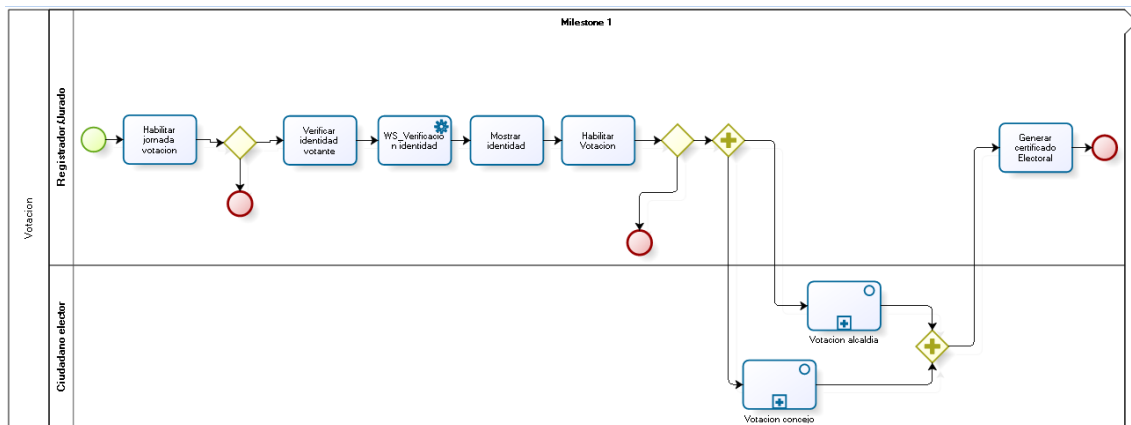


Figura 15. Modelo BPMN del subproceso de votación

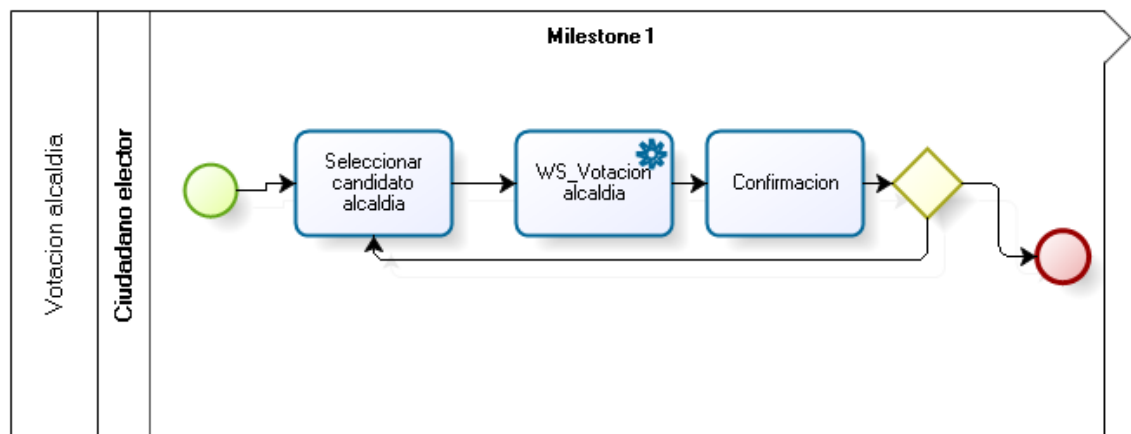


Figura 16. Modelo BPMN del subproceso de votación alcaldía

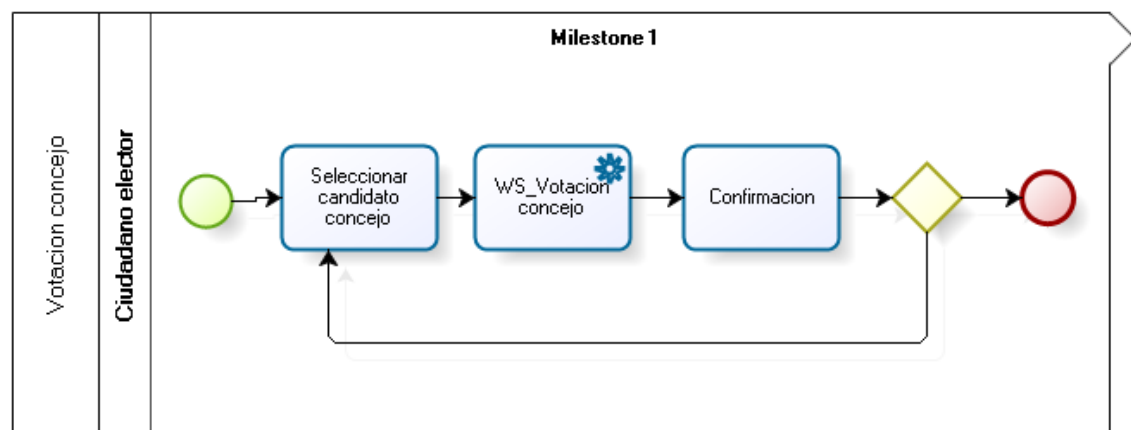


Figura 17. Modelo BPMN del subproceso de votación concejo

9.5.2. Imágenes del prototipo

The image displays three screenshots of a web application prototype for Bizagi, running in Internet Explorer. The browser's address bar shows the URL [http://muser-978afce5\(Evoting\)](http://muser-978afce5(Evoting)).

Screen 1: Bienvenido a Bizagi
This screen shows the login interface. It features the Bizagi logo at the top. Below it, a message says "Bienvenido a Bizagi" and "Digite sus datos para ingresar a Bizagi". There is a "Usuario" dropdown menu with options: "admon", "admon", "Administrador", "Auxiliar", "Votante", "Testigo", and "registrado2". An "Ingresar" button is located below the menu.

Screen 2: App - Evoting - Solicitud inscripción candidato
This screen shows the candidate registration form. It includes a "Cedula" field with the value "1098626261". Below the field are "Guardar" and "Siguiente >>" buttons. A summary box on the left contains the following information:
Número Alta: 1001
Fecha de Solución: 29/11/2011
Creado por: admin
Encargado Actual: admin
Encargados Actuales de Eventos: admin
At the bottom, there is a "Powered by bizagi" logo and a "Cedula" label next to a small input field.

Screen 3: App - Evoting - Verificar requisitos
This screen shows the requirements verification form. It includes fields for "Cedula" (1098626261), "Nombre" (daniel), and "Apellido" (caceres). There are radio buttons for "Avala" (selected) and "No", and another set for "CertResidencia" (selected) and "No". Below these are "Guardar" and "Siguiente >>" buttons. The same summary box from the previous screen is present on the left. The "Powered by bizagi" logo and a "CertResidencia" label next to a small input field are at the bottom.

Screen 4: App - Evoting - Confirmar registro
This screen shows the registration confirmation form. It includes fields for "FechaInscripcion" (lunes, 29 de noviembre de 2011), "Cedula" (1098626261), "DirResidencia" (Calle 1 # 1 - 05), "Telefono" (4578947), "Puesto" (Conservador), "NumeroCandidato" (1), "Municipio" (Cundinamarca - Choachi), "Cargo" (Alcalde), "URLugVolCand" (Bogota), and "Confirm" (selected) and "No" radio buttons. Below these are "Guardar" and "Siguiente >>" buttons. The same summary box from the previous screen is present on the left. The "Powered by bizagi" logo and a "Confirm" label next to a small input field are at the bottom.

App - Evoting - Entregar certificado inscrip cand

FechaInscripcion:	lunes, 28 de noviembre de 2011
Cedula:	1098626261
Partido:	Conservador
NumeroCandidato:	1
Municipio:	Cundinamarca - Choachi
Cargo:	Alcaldia

Guardar **Siguiente >>**

Número Alta:	1001
Fecha de Solución:	28/11/2011
Creado por:	admon
Encargado Actual:	admon
Encargados Actuales de Eventos:	

App - Evoting - Registrar candidato

Cedula:	1098626261
Dirección:	Calle 1 # 1- 05
Telefono:	4378547
Partido:	Conservador
NumeroCandidato:	1
Departamento:	Cundinamarca
Municipio:	Choachi
Cargo:	Alcaldia
URLug/VotCand:	

Guardar **Siguiente >>**

Número Alta:	1001
Fecha de Solución:	28/11/2011
Creado por:	admon
Encargado Actual:	admon
Encargados Actuales de Eventos:	

App - Evoting - Habilitar inscripción

HabilitarCand: ☐ SI ☐ No

Guardar **Siguiente >>**

Número Alta:	1001
Fecha de Solución:	28/11/2011
Creado por:	admon
Encargado Actual:	admon
Encargados Actuales de Eventos:	

Powered by **bizagi**

App - Evoting - Seleccionar candidato alcaldia

Departamento:	Cundinamarca
Municipio:	Choachi
PartidoAlc:	Conservador
CandidatosAlc:	Charles Puyol

Guardar **Siguiente >>**

Número Alta:	1001
Fecha de Solución:	28/11/2011
Creado por:	admon
Encargado Actual:	admon
Encargados Actuales de Eventos:	

ACTIVIDADES
[Seleccionar candidato concejo](#)

Powered by **bizagi**

Por favor llene los campos de la forma.

9.6. Validación según TAM

El Modelo de Aceptación de la Tecnología (Technology Acceptance Model - TAM) de Davis [21] es una teoría de los sistemas de información que modela cómo los usuarios llegan a aceptar y utilizar una tecnología. El modelo sugiere que cuando a los usuarios se les presenta una nueva tecnología, una serie de factores influyen en su decisión sobre cómo y cuándo la van a utilizar, dicho de otra forma, sirven de base para determinar las actitudes enfocadas al uso del sistema. Estos factores son:

- PU (Perceived usefulness, Utilidad Percibida): “*el grado en que una persona cree que el uso de un determinado sistema mejora su rendimiento en el trabajo*”.
- FUP (Perceived ease-of-use, Percepción de facilidad de uso): “*el grado en que una persona cree que utilizando un sistema en particular, podrá liberarse del esfuerzo que le conlleva realizar un trabajo*”. [22]

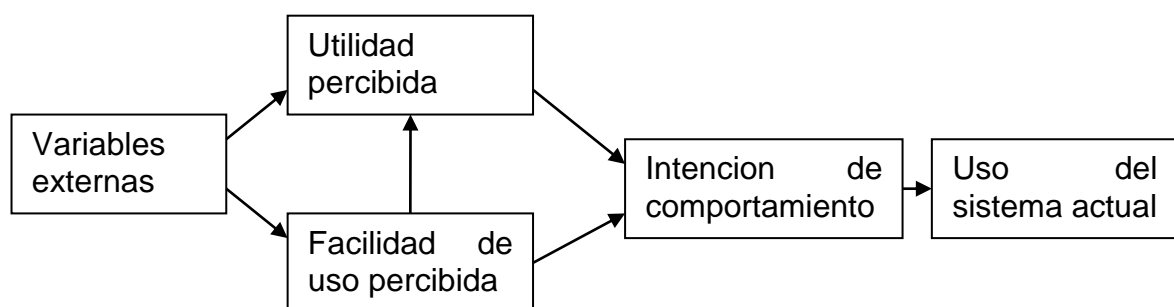


Figura 18. Modelo de Aceptación de la tecnología. Fuente [22]

Para este proyecto, el usuario elegirá entre dos sistemas con funciones idénticas: el sistema de registro tradicional y votación de la registraduría municipal de Choachí y el sistema propuesto en el proyecto representado mediante el prototipo funcional de interfaces de usuario de inscripción de electores y candidatos.

La forma de realizar esta validación se dio mediante el siguiente formulario de encuesta que se presentó al registrador municipal, al personero municipal y a un ciudadano al azar. Se tomó como guía el tipo de encuesta realizado por Orantes [23], con respectivos ajustes para el entorno en el cual se va a aplicar, además de controlar las opciones de votación de 7 opciones como maneja orantes a 5 (totalmente de acuerdo - 5, de acuerdo - 4, ni acuerdo ni desacuerdo - 3, en desacuerdo - 2, totalmente en desacuerdo - 1):

Facilidad de Uso

- Preg.1: ¿Es fácil aprender a operar el nuevo sistema?
- Preg.2: ¿Es fácil acceder al sistema para hacer lo que deseo?
- Preg.3: ¿Es fácil aumentar mi experiencia gracias al uso del nuevo sistema?
- Preg.4: ¿El nuevo sistema es fácil de utilizar?
- Preg.5: ¿Sería fácil llegar a ser un experto en el manejo del sistema?

Utilidad Percibida

- Preg.6: ¿El uso del nuevo sistema me ayuda a hacer las tareas más rápido?
- Preg.7: ¿El uso del nuevo sistema mejoraría el funcionamiento de mi trabajo dentro de la registraduría?
- Preg.8: ¿El uso del nuevo sistema aumenta mi productividad dentro de la registraduría?
- Preg.9: ¿El uso del nuevo sistema incrementa mi efectividad dentro de la registraduría?
- Preg.10: ¿El nuevo sistema es de utilidad dentro de la registraduría?

Actitud hacia el Uso.

- Preg.11: ¿Usar el nuevo sistema es una buena idea?
- Preg.12: ¿Usar el nuevo sistema es una inteligente idea?
- Preg.13: ¿Me gusta la idea de usar el nuevo sistema?
- Preg.14: ¿Usar el nuevo sistema me parece (placentero/no-placentero)?

Hay que aclarar que de las 14 preguntas, solo el registrador debía contestarlas todas. Las otras dos personas al solo interactuar como quienes recibían la inscripción y como votantes respondieron solo las preguntas de facilidad de uso y de actitud hacia el uso.

El resultado por parte del registrador fue:

Pregunta 1	5
Pregunta 2	4
Pregunta 3	5
Pregunta 4	5
Pregunta 5	4
Pregunta 6	5
Pregunta 7	5
Pregunta 8	5
Pregunta 9	4
Pregunta 10	4
Pregunta 11	4

Pregunta 12	4
Pregunta 13	4
Pregunta 14	3

El resultado del Personero (Diego Pardo) y de una votante de las pasadas elecciones en el municipio (Benedicta Rincón) son:

Diego Pardo		Benedicta Rincón	
Pregunta 1	5	Pregunta 1	5
Pregunta 2	4	Pregunta 2	4
Pregunta 3	5	Pregunta 3	5
Pregunta 4	5	Pregunta 4	5
Pregunta 5	4	Pregunta 5	4
Pregunta 6	5	Pregunta 6	5
Pregunta 7	5	Pregunta 7	5
Pregunta 8	5	Pregunta 8	5
Pregunta 9	4	Pregunta 9	4
Pregunta 10	4	Pregunta 10	4
Pregunta 11	4	Pregunta 11	4
Pregunta 12	4	Pregunta 12	4
Pregunta 13	4	Pregunta 13	4
Pregunta 14	3	Pregunta 14	3

Las validaciones realizadas para toda la arquitectura nos llevan a deducir que aunque esta propuesta de voto electrónica dentro de esquemas complejos o mundiales sea una fase inicial o de estudios guías, se presenta como el camino o pauta para poder implementar una solución tecnológica manteniendo la coherencia de la solución con el problema desde donde se origina ya que en Colombia no se ha presentado modelo de arquitectura similar (solo pilotos sobre aplicaciones). Factores como la comprensión de la realidad colombiana y de su ente gestor de las elecciones, así como la presentación de un prototipo y un nivel de aceptación aceptable del mismo por parte de los futuros usuarios permite entender que este proyecto sea visto una vía de innovación real no muy lejana para Colombia

10. Bibliografía

- [1] Harmon, P. *"Business Process Methodologies"*. BPTrends - Business Process Trends. Volumen 5. Number 20. 2007
- [2] Forster, F. *"The Idea behind Business Process Improvement: Toward a Business Process Improvement Pattern Framework"*. BPTrends – Business Process Trends. 2006
- [3] Kim, D., Kim, M., Kim, H. *"Dynamic Business Process Management based on Process Change Patterns"*. International Conference on Convergence Information Technology. Pp: 1154 – 1161. 2007.

- [4] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. *"Pattern-Oriented Software Architecture Volume 1: A System of Patterns"*. Wiley. Pp: 31. 1996
- [5] United States Treasury Department. *"US treasury architecture development guidance-tadg"*. Section 7. 1997.
- [6] Monday, P. *"Web services patterns: java edition"*. Apress. 1 edition. ISBN-10: 1590590848. 2003
- [7] Lawrence Pfleeger, S. *"Software Engineering Theory and practice"*. Beohm McCall ISO Model. Prentice Hall. 2001.
- [8] Boehm, B.W., Brown, J.R., Lipow, M. *"Quantitative evaluation of software quality"*. International Conference on Software Engineering, Proceedings of the 2nd international conference on Software engineering(2nd). Pp: 592-605. 1976.
- [9] Grady, R., Caswell, D. *"Software Metrics: Establishing a Company-wide Program"*. Prentice Hall. Pp: 159. 1987. ISBN 0138218447.
- [10] ISO/IEC 9126-1, Software engineering – product quality – Part 1: Quality Model, first ed.: June 15th. 2001
- [11] Dromey, R.G. *"A model for software product quality"*. IEEE Transactions on Software Engineering 21. Volumen 2. 1995.
- [12] Fitzpatrick, R. *"Software quality definitions and strategic issues"*. Staffordshire University, 1996.
- [13] Neil, M., Fenton, N. *"Predicting software quality using bayesian belief networks"*. NASA/Goddard Space Flight Centre. December 1996.
- [14] Khosravi, K., Gueheneuc, Y.G. *"A Quality Model for Design Patterns"*. Technical Report 1249. University of Montreal. 2004.
- [15] Gamma, E., Helm, R., Johnson, R., Vlissides, J. *"Design Patterns Elements of Reusable Object-Oriented Software"*. Addison-Wesley Pub Co. 1995.
- [16] Nagappan, R., Skoczylas, R., Sriganesh, R.P. *"Developing Java Web Service: Architecting and developing secure web services using java"*. Wiley Publishing. ISBN 0-471-23640-3. 2003.
- [17] Lakshminarayanan, S. *"Oracle Web Services Manager: securing your web services"*. Packt publishing. First published. ISBN 978-1-847193-83-4. 2008
- [18] Bien, A. *"Real World Java EE Patterns: Rethinking Best Practices"*. Lulu.com. ISBN-10: 0557078326. 2009
- [19] Clements, P., Kazman, R., Klein, M. *"Evaluating software Architectures: methods and case studies"*. Addison-Wesley Professional. 1 edition. 2001
- [20] Clements, P., Kazman, R., Klein, M. *"ATAM: Method for architecture evaluation – Technical Report"*. Software Engineering Institute. Carnegie Mellon. 2000
- [21] Davis, F.D. *"User acceptance of information technology: system characteristics, user perceptions and behavioural impacts"*. International Journal of Man-Machine Studies 38, pp. 475-487. 1993
- [22] Davis, F.; Bagozzi, R.; and Warshaw, R. *"User Acceptance of Computer Technology: A Comparison of Two Theoretical Models"*. Management Science, Volume 35, 1989, pp. 982-1003. 1989
- [23] Orantes Jiménez, S.D. *"Viabilidad del Modelo de Aceptación de la Tecnología en las empresas mexicanas. Una aproximación a las actitudes y percepciones de los usuarios de las tecnologías de la información"*. Revista Digital Universitaria. Volumen 12. Número 1. ISSN: 1067-6079. 2011