**Sistema de Votaciones para ONG**

**(SAD) Software Architecture Document**

**Versión 1.0**

* **Tabla de Contenidos**

Contenido

[**● Tabla de Contenidos 0**](#_heading=h.gkp737h9219k)

[**● Identificación de Documento 2**](#_heading=h.kvyhq8ffddmh)

[**● Historia de cambios 2**](#_heading=h.42sj4ucaubjt)

[**● Índice de figuras 3**](#_heading=h.3znysh7)

[**● Introducción 4**](#_heading=h.2s8eyo1)

[**● Alcance 4**](#_heading=h.zet794er9cuo)

[**● Referencias 4**](#_heading=h.3rdcrjn)

[**● Arquitectura de Software 4**](#_heading=h.26in1rg)

[**● Objetivos y Restricciones de la Arquitectura 4**](#_heading=h.lnxbz9)

[**● Diagramas 4**](#_heading=h.35nkun2)

[**● Tamaño y desempeño 4**](#_heading=h.xh4j8fds8gvf)

* Identificación de Documento

| **Identificación** |  |
| --- | --- |
| **Proyecto** | Sistema de Votaciones para ONG |
| **Versión** | 1.0 |

| **Documento mantenido por** |  |
| --- | --- |
| **Fecha de última revisión** | 16/11/2024 |
| **Fecha de próxima revisión** | 20/11/2024 |

| **Documento aprobado por** | Mauricio Cabrera |
| --- | --- |
| **Fecha de última aprobación** | 16/11/2024 |

* **Historia de cambios**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 16/11/2024 | 1.0 | Creacion y Planificacion del Documento DAS | Bastian Barrera |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

[**1** **Introducción 3**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.gjdgxs)

[1.1 Contexto del Problema 3](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.30j0zll)

[1.2 Propósito](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.1fob9te)

[1.3 Referencias 3](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.tyjcwt)

[**2** **Representación de la Arquitectura 5**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.3whwml4)

[2.1 Representación 5](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.1t3h5sf)

[**3** **Metas y Restricciones de la Arquitectura 6**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.4d34og8)

[A continuación se revisan las metas y restricciones de la arquitectura. 6](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.2s8eyo1)

[3.1 Metas de la arquitectura 6](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.17dp8vu)

[3.3 Otros antecedentes y consideraciones 6](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.26in1rg)

[**4** **Vista de Casos de Uso y Escenarios de Calidad 8**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.lnxbz9)

[4.1 Modelo de Casos de Uso 8](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.35nkun2)

[4.2 Especificación de Casos de Uso Relevantes 8](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.1ksv4uv)

[4.3 Especificación de los Escenarios de Calidad Relevantes 9](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.44sinio)

[**Vista Lógica 11**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.2jxsxqh)

[4.4 Parte Estructural 11](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.z337ya)

[4.5 Parte Dinámica 11](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.3j2qqm3)

[**5** **Vista de Procesos 13**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.1y810tw)

[**6** **Vista de Implementación 14**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.4i7ojhp)

[**7** **Vista de Despliegue 15**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.2xcytpi)

[**8** **Informe de errores 16**](https://docs.google.com/document/d/1GjB6KZ4qIR5ZST83v_B5J18X5irdeFw7/edit#heading=h.1ci93xb)

1. **Introducción**
   1. **Contexto del Problema**

Actualmente en Chile y a nivel mundial no existe fiabilidad en los sistemas de Votaciones actuales debido a falencias, usurpación de datos y manipulación en votos electorales , es por esto, que AMB.corp presenta una propuesta novedosa que rompe los estándares y da a conocer un Sistema de Votaciones realizado en Salesforce con un lenguaje de programación en APEX, el cual promueve la fiabilidad, seguridad y confidencialidad a los datos e información que se manipula al estar principalmente enfocado hacia las Organizaciones No Gubernamentales, las cuales, requieren un sistema económico, fiable y seguro dentro de sus sistemas para ejecutar las tareas y acciones dentro de las votaciones electorales.

Debido al desarrollo de este sistema no tan solo podría ser utilizado para las Organizaciones no Gubernamentales sino para otros fines como servicios públicos, servicios de venta, E-commerce dando la oportunidad de una innovación versátil que cumple los requerimientos y necesidades de nuestros clientes.

* 1. **Propósito**

Como propósito principal de nuestro proyecto es cubrir la necesidad de fiabilidad en los procesos electorales dentro de las Organizaciones No Gubernamentales, es por ello que se desarrolla este sistema novedoso utilizando nuevas tecnologías como Apex con VisualForce además de la plataforma en la nube Salesforce para crear un sistema fiable, seguro y confidencial para cubrir la inseguridad de los votantes y los futuros usuarios que utilicen esta plataforma.

* 1. **Referencias**

A continuación se listan las referencias a otros documentos:

* **Acta de constitución**
* **Documento ERS**

* 1. **Representación**

La arquitectura del Proyecto Delivery está representada siguiendo el enfoque del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Casos de Uso y Escenarios de Calidad**: Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.
* **Vista de Metas y Restricciones**: Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.
* **Vista Lógica**: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. Usa el view type Módulos para representar la estructura lógica y el view type Componentes y Conectores para representar el comportamiento.
* **Vista de Procesos**: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.
* **Vista de Implementación**: Describe los componentes de deployment construidos y sus dependencias.

1. **Introducción**

A continuación se revisan las metas y restricciones de la arquitectura.

**2.1 Metas de la arquitectura**

De acuerdo a las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

* **Aprendibilidad**
* Estabilidad
* Fiabilidad
* Confidencialidad

**2.2 Restricciones de la Arquitectura**

Existen restricciones que han sido levantadas con los stakeholders, las cuales se presentan a continuación:

* **Tiempo de construcción**: se cuenta con un plazo estrecho de tiempo para su construcción, 1 semana según la planificación dando así un plazo acorde al tiempo entregado.
* **Infraestructura**: se cuenta con el desarrollo del sistema con Salesforce, una empresa con soporte en la nube el cual proporciona fiabilidad, confidencialidad y eficacia en los procesos del sistema.
* **Otros componentes de software**: Se necesita el Hardware para llevar a cabo el desarrollo del proyecto que serian destinados un notebook, Licencias de Salesforce y el conocimiento y la expertiz del equipo de trabajo.

**3.1 Otros antecedentes y consideraciones**

La empresa desarrolladora cuenta con un framework que considera los siguientes componentes que permiten satisfacer los requerimientos arquitectónicos:

* Framework , con esto se soporta la encapsulación y modularización de componentes para facilitar la mantenibilidad del sistema. Asimismo, privilegia el performance en tiempo de ejecución dado que es un framework liviano.
* Framework de seguridad, con esto se soporta la meta de seguridad.

.

1. **Alcance**

**Especificación de Casos de Uso Relevantes**

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

* Su implementación implica mejor factibilidad, gestión y valor de seguridad a las ONGs
* Incluye muchos conceptos y relaciones del dominio.
* Incluye posibles escenarios críticos de calidad.

A continuación se listan los casos de uso relevantes:

| **Código** | **Nombre** | **Actores** | **Prioridad** |
| --- | --- | --- | --- |
| CU-001-001 | Cada usuario puede realizar una votación | AMB.Corp | alta |
| CU-002-002 | Se generan registro de votaciones | AMB.Corp | alta |
| CU-003-003 | Generan reportes en tiempo real | AMB.Corp | alta |
| CU-004-004 | Gestión de usuarios para la plataforma | AMB.Corp | alta |
| CU-005-005 | Registro de Usuarios y Votantes a la plataforma | AMB.Corp | alta |
| CU-006-006 | Configuración de notificaciones a correo electrónico | AMB.Corp | alta |
|  |  |  |  |

1. **Referencias**

De acuerdo a las referencias que se utilizaron para crear el sistema y la documentación pertinente, tenemos por entendido que se utilizará Salesforce como plataforma en la nube, la cual es la adecuada para cumplir con las funcionalidades solicitadas. Además de eso se utilizarán como lenguaje Apex y VisualForce garantizando eficiencia en el sistema. Además se utilizará la metodología tradicional debido a que se utilizaran procesos lineales y con fases ya definidas. Junto a esto mejoraremos las prácticas en automatización, gestión de proyectos y regulaciones sobre votos a ONGs.

1. **Arquitectura de Software**

En esta sección se dará a conocer la arquitectura que se utilizará dentro del desarrollo de software para las Organizaciones No Gubernamentales. En relación al contexto se va a utilizar la siguiente arquitectura , la cual representa diferentes vistas obteniendo perspectivas del sistema, creando un diseño intuitivo, garantizando una vista general del sistema. A continuación se da a conocer la estructura de la arquitectura mencionando los procesos y las funcionalidades del sistema.

Arquitectura basada en Salesforce (Modelo SaaS) en este caso, utilizamos Salesforce como la plataforma principal para gestionar todos los aspectos del sistema de votación. Salesforce ofrece funcionalidades robustas y servicios gratuitos para organizaciones sin fines de lucro. Aprovechando su infraestructura y licencias gratuitas, este modelo reduce los costos de implementación.

Componentes: Salesforce: El sistema central para gestionar usuarios, votaciones y resultados. Salesforce Lightning para la interfaz de usuario. Apex para la lógica del negocio (gestión de votos, validación de usuarios, etc.). Visualforce para la creación de formularios de votación personalizados. Base de Datos Salesforce: Gestión de votaciones, registros de usuarios, seguridad de datos y resultados. Mailjet/Mailchimp API: Integración para enviar notificaciones o recordatorios por correo electrónico a los participantes. Autenticación y Seguridad: Uso de OAuth 2.0 y Salesforce Identity para garantizar la privacidad y seguridad del voto. Licencias gratuitas de Salesforce para ONGs. Ventajas: Plataforma robusta y segura. Fácil integración con otros sistemas. Licencias gratuitas para ONGs. Capacidad de escalar sin preocuparse por la infraestructura. Desventajas: Dependencia de la infraestructura de Salesforce. Puede requerir capacitación adicional para la personalización.

**Sistema de Votaciones en Salesforce (AMB.Corp)**

* Entorno central donde se desarrolla una solución a la necesidad de obtener un sistema confiable, confidencial y seguro para los votantes basado en una plataforma en la nube.
* El sistema garantiza seguridad, accesibilidad y disponibilidad en cualquier momento.

**Backend (Salesforce y Apex)**

El entorno de procesamientos está en Salesforce, utilizando Apex como lenguaje de programación para gestionar la lógica del negocio entre ellas:

### Validar Rut

* Registro de votos y Futuras mantenciones
* Control del estado de las votaciones
* Además de mediante el Backend administrar la vista de administrador, donde se configuran y monitorizan los estados de las votaciones.

#### **Front-end (VisualForce)**

#### La interfaz de VisualForce permite construir interfaces de usuario, mejorando la compatibilidad con diferentes dispositivos, mejorando la accesibilidad en la plataforma.

* Permite que los usuarios puedan ver sus votaciones disponibles, emitir su voto y consultar los resultados en tiempo real.

### **Base de datos en Salesforce**

La base de datos está integrada a Salesforce y utiliza un modelo orientado a objetos en la cual se almacena información clave como datos de los votantes tales como rut, correo electrónico, además de votaciones y sus estados y votos registrados de manera cifrada para garantizar privacidad.

### **Flujos de trabajo y automatización**

Salesforce facilita los procesos clave como el envío de correos electrónicos con URLs únicas para acceder a las votaciones. Además de generación automática de reportes y notificación de finalización de votantes.

### 

### **Herramientas de Desarrollo (Eclipse)**

La herramienta de desarrollo Eclipse es utilizada para escribir y gestionar el código en Apex y desarrollar páginas en VisualForce, integrando mediante estas herramientas y tecnologías las funcionalidades requeridas por el sistema.

Finalizando se da a conocer las diferentes herramientas, tecnologías y métodos en nuestra arquitectura de software para mantener la fiabilidad del sistema otorgando funcionalidades que van de acorde a los requerimientos del cliente en este caso las organizaciones no gubernamentales, es por esto que los futuros usuarios (votantes) van a interactuar con el sistema mediante diferentes dispositivos, asegurando un diseño responsivo y amigable para usuarios con habilidades técnicas limitadas.

1. **Objetivos y Restricciones de la Arquitectura**

Para llevar a cabo la arquitectura se tienen que tomar en cuenta los objetivos y restricciones que tenemos para llevar a cabo el desarrollo del sistema con esta arquitectura. Es por esto que se mencionan para tener en cuenta objetivos claros y las restricciones que se deben tomar en cuenta al momento de poner el marcha el sistema de votaciones.

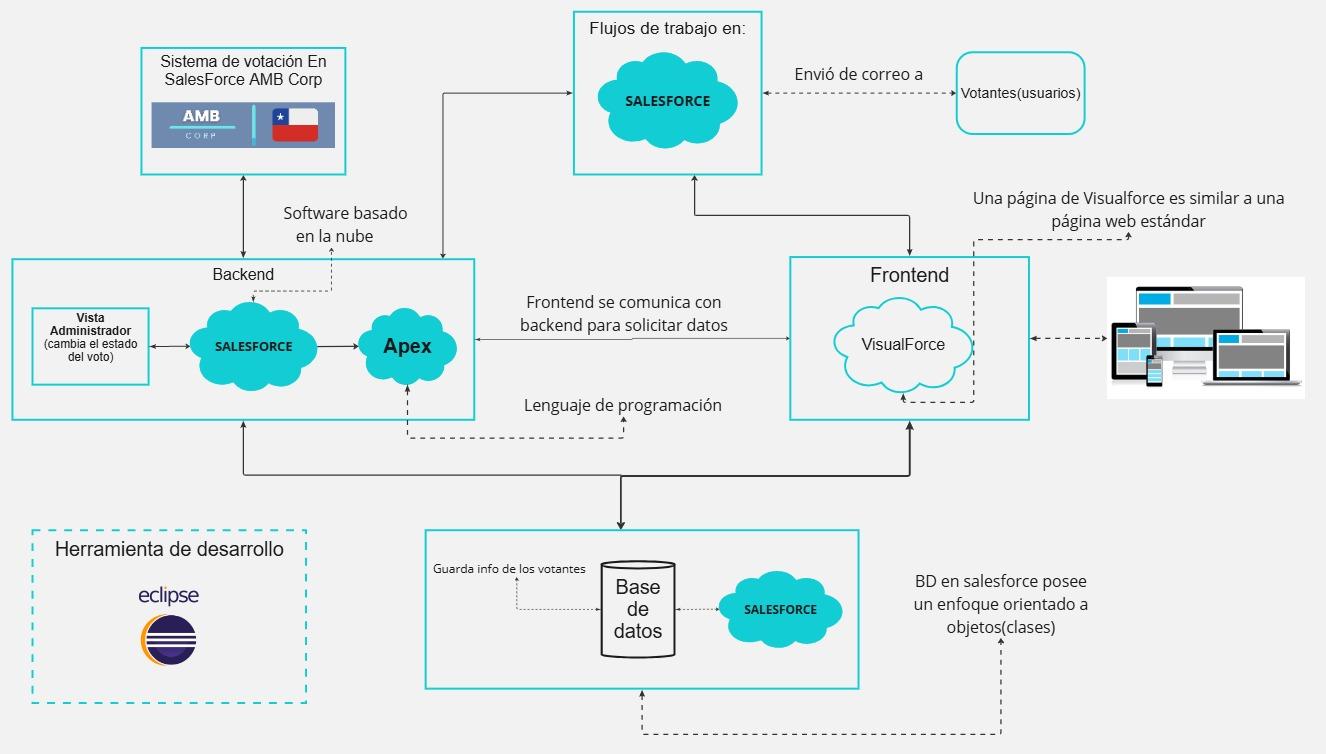
Como objetivos principales de la arquitectura es que facilita la gestión del proyecto aportando escalabilidad, ofreciendo soporte, garantizando transparencia y seguridad. En cuanto a las restricciones de la arquitectura es que las dependencias están en Salesforce por lo que el sistema permanecerá desplegado en la plataforma, teniendo limitaciones de control, además del tiempo de implementación, recursos disponibles y capacidades técnicas para llevar a cabo el desarrollo eficaz de la aplicación. Por lo tanto, se mencionan estas características para saber el funcionamiento, beneficios y riesgos que otorga trabajar junto a esta arquitectura que resulta beneficiosa, ya que, contamos con anterioridad los requerimientos por lo tanto nuestro trabajo es lineal y continuo.

1. **Diagrama**

En esta sección se da a conocer el diagrama de arquitectura el cual muestra la estructura y funcionamiento del sistema de votaciones implementado en Salesforce, diseñado principalmente para las organizaciones no gubernamentales. Este sistema utilizará de Frontend basado en VisualForce, proporcionando una interfaz amigable, responsive y accesible desde cualquier dispositivo para permitir que los usuarios realicen acciones de validar su identidad con su RUT, visualizar votaciones disponibles, emitir votos y consultar resultados en tiempo real.

Además el backend está compuesto por Salesforce y programado en Apex, gestionando la lógica del negocio, incluyendo autenticación de usuarios, automatización de procesos como el envío de correos electrónicos con enlaces de votación, registro de votos con encriptación y generando reportes.

La base de datos, integrada con Salesforce, utiliza un modelo orientado a objetos para almacenar y proteger los datos de los usuarios y del sistema, garantizando la privacidad e integridad de los datos. El sistema deberá ser escalable y accesible, soportando votaciones masivas y garantizando un tiempo de respuesta eficaz, ayudando a la toma de decisiones de las ONGs. Por último, el desarrollo se llevó a cabo en Eclipse como herramienta principal, asegurando un código limpio y estructurado proporcionando una solución robusta, segura, eficiente y acorde a las necesidades de las organizaciones no gubernamentales.

****

1. **Tamaño y Desempeño**

En relación a la siguiente sección se da por entendido que el tamaño del sistema permite estimar los recursos requeridos para implementar el sistema de votaciones todo esto conociendo los datos, código y componentes del software.

* Volumen esperado de datos: Dentro de las funcionalidades del sistema se espera que se puedan realizar 1.000 registros iniciales, escalables, además de votaciones simultáneas de 50-100 activas simultáneamente en fases iniciales, además de resultados y reportes.
* Tamaño por registro: Dentro del tamaño que pueden tener los registros de los usuarios se mantiene en 2kb, opciones y resultados son 1kb por opción de voto.
* Código del sistema: Se da a conocer los lenguajes que se utilizaron para código VisualForce, Controladores APEX y Test unitarios

Además se tiene que evaluar el desempeño del sistema para así verificar que el sistema cumpla con las expectativas en tiempo de respuesta, uso de recursos y escalabilidad. En este sentido mencionamos características que ayudan a medir el desempeño del sistema.

* Tiempo de respuesta: Se da por entendido que el tiempo de respuesta es de 2 segundos para operaciones críticas como el registro de votos y generación de reportes en tiempo real
* Alta capacidad de interacciones con el sistema: se espera hasta 50 usuarios simultáneamente y una escalabilidad a futuro de 500 usuarios concurrentes.
* Uso de recursos: El uso principal de recursos es mayormente por parte de la cpu y memoria para procesamiento de Salesforce.
* Tolerancia a fallos: El sistema estará desplegado la mayor parte del tiempo tratando así de obtener una disponibilidad del 100%.

En relación a lo antes mencionado se da por entendido que se tiene que dar conocimiento al tamaño del sistema para saber el tamaño de las ocurrencias dentro del sistema y desempeño para orientar a los usuarios y votantes a usar el sistema adecuadamente.