PLAN DE ACTUALIZACIONES

Servicio para la actualización y mantenimiento del software de escrutinios, que utiliza el Consejo Nacional Electoral, para los procesos electorales a llevarse a cabo en las vigencias 2025 y 2026

Diciembre 2025

INDRA COLOMBIA S.A.S.

Toda la información contenida en el presente documento y sus anexos tiene carácter confidencial, y sólo puede ser utilizada con el fin de ser evaluada por el destinatario (sea cliente, proveedor, colaborador, partners, etc.) de la misma y a los solos efectos de conducir los tratos comerciales, o de otra naturaleza, que motivan el envío del documento (en lo sucesivo, el "Propósito").

La información aquí presentada es propiedad de la Oferente que suscribe el presente documento, y es constitutiva de secreto empresarial (también denominado en determinadas jurisdicciones, secreto comercial), y además, puede estar protegida por derechos de autor, derechos afines, patente, modelo de utilidad y/o diseño industrial por lo que queda terminantemente prohibida su divulgación y/o transmisión a terceros sin el permiso previo, expreso y por escrito.

Se limitará al máximo el acceso a la información confidencial por parte del personal del destinatario de la misma, o del personal de aquellos terceros a los que se haya autorizado a acceder a la información confidencial, limitándose únicamente a aquellas personas cuyo acceso resulte estrictamente necesario, y debiendo el destinatario de la información confidencial garantizar que informa a dichas personas del carácter confidencial y propietario de la información así como del Propósito, asegurando que dicho personal trata la información confidencial única y exclusivamente para el Propósito, y absteniéndose de toda divulgación.

Una vez finalizado o concluido el Propósito, el cliente deberá restituir toda la información confidencial sin conservar ninguna copia de la misma, no pudiendo utilizarla de ninguna manera ni para ningún otro fin, salvo autorización previa, expresa y por escrito.

Copyright. Todos los derechos reservados. Colombia

Índice

1 Introducción 5

2 Descripción del Plan 6

2.1 Fase de revisión funcional 6

2.2 Fase de revisión de diseño gráfico 6

2.3 Fase de mantenimiento preventivo 7

2.4 Fase de auditoría 7

2.5 Fase de planes de pruebas 8

2.6 Fase de Mantenimiento Correctivo 8

2.7 Fase de Entregas 9

3 Mecanismos de control de calidad y validación 10

3.1 Metodología de Calidad 10

3.1.1 Integración temprana del equipo de QA en el ciclo de vida del proyecto 10

3.1.2 Definición y ejecución de procedimientos de prueba 10

3.1.3 Validación automatizada de datos procesados 10

3.1.4 Validación web multiplataforma y multinavegador 11

3.1.5 Pruebas de accesibilidad digital con equipo especializado independiente 11

3.1.6 Control de calidad tras cada actualización 11

3.2 Herramientas para la gestión de pruebas 11

3.2.1 Uso de Qase en el proyecto 12

3.2.2 Uso de Microsoft Planner en el proyecto 12

3.2.3 Integración operativa entre Qase y Planner 13

3.2.4 Beneficios para la gestión del proyecto 13

3.3 Herramientas de automatización 14

3.3.1 Selenium 14

3.3.2 Appium 15

3.3.3 BrowserStack 15

3.3.4 Uso conjunto de las herramientas en el proyecto 16

3.3.5 Beneficios directos para la gestión del proyecto 16

3.4 Herramientas de accesibilidad 17

3.4.1 Herramientas comunes 17

3.4.2 Herramientas para web 17

3.4.3 Herramientas para dispositivos móviles 17

4 Metodologías de desarrollo 18

4.1 Metodología ágil 19
4.2 Ciclo de vida de desarrollo de software seguro 20
4.3 Control de versiones 21
4.4 Seguridad del código 21
4.5 Integración continua 22

1 Introducción

En este documento se describe el plan de actualizaciones que desarrollará Indra para mantenimiento del software de escrutinio de la Comisión Escrutadora Nacional del CNE, así como del sistema de control y seguimiento al escrutinio nacional disponible en la web, durante 2026.

A continuación, se detallan las fases principales del proceso de actualización y mantenimiento, así como en cada fase se describen las metodologías y mejores prácticas que utiliza el equipo de desarrollo y de gestión del proyecto, así como se describen las herramientas tecnológicas y el uso combinado que hace el equipo para lograr resultados de la mayor calidad en el producto/software liberado.

2 Descripción del Plan

2.1 Fase de revisión funcional

En esta primera fase del proyecto, se enumerarán de las funcionalidades de los componentes del software de Escrutinio Nacional, separado por los diferentes módulos que lo componen. Una vez hecho esto se realizará el siguiente análisis:

- Con la experiencia de la ejecución de los procesos electorales de 2022, se revisará si es posible aplicar alguna mejora operativa al sistema.
- Después, se analizará el listado de tareas a realizar en la fase de mantenimiento preventivo del proyecto.

Este análisis será compartido con la Registraduría para su revisión, donde también puede incluir necesidades o evolutivos detectados en los procesos electorales anteriores.

Además, la entidad debe de encargarse de la entrega de documentación necesaria para crear un escenario de datos en cada una de las elecciones a realizar en 2026.

- DIVIPOL.
- Archivos básicos con listado de candidaturas y curules.
- Formatos de E-14, E-24, E-26y del AGE Nacional, a fin de determinar las diferencias con lo realizado en 2022.

Como resultado de esta fase se debería obtener una relación de cambios funcionales, que encajen en tiempo y forma dadas las fechas de las elecciones en 2026.

2.2 Fase de revisión de diseño gráfico

Con el equipo de diseño de Indra, se repasa el diseño gráfico de las aplicaciones web de la Comisión Nacional, para realizar los ajustes necesarios en materia gráfica.

Este proceso incluye una revisión exhaustiva de todos los elementos visuales presentes en las aplicaciones, asegurando que cumplan con los lineamientos de identidad institucional y las mejores prácticas de diseño centrado en el usuario. Las actividades abarcan:

- Validación de la paleta cromática para garantizar coherencia con la imagen corporativa de las elecciones 2026, y también teniendo en cuenta el contraste de colores para cumplir con la legislación Colombiana en materia de accesibilidad y con los estándares internacionales (WCAG).
- Revisión de tipografías y jerarquías visuales, optimizando la legibilidad en distintos dispositivos y resoluciones.
- Ajuste de la disposición de componentes gráficos, buscando una experiencia intuitiva y consistente en todas las pantallas.
- Actualización de diseño responsive en el módulo de seguimiento web por internet, asegurando que la interfaz se adapte correctamente a diferentes tamaños de pantalla. El seguimiento web público podrá ser visualizado correctamente en terminales móviles con los sistemas operativos principales (iOS y Android), y en terminales fijos con los navegadores más extendidos (Edge, Firefox, Chrome y Safari).

- Integración de mejoras en la experiencia de usuario (UX), priorizando la claridad, simplicidad y eficiencia en la navegación.

Se entregará a la Registraduría un documento con el diseño actualizado para abordar las elecciones de 2026.

2.3 Fase de mantenimiento preventivo

En esta fase se realizarán todas las tareas de desarrollo asociadas a la actualización del sistema, tanto aquellas mejoras que se decidieran realizar en la fase de revisión funcional, además de las actualizaciones recomendadas sobre los diferentes componentes como actualizaciones de versiones de librerías utilizadas o mejoras asociadas al performance y rendimiento.

A continuación, se enumeran alguna de las tareas a realizar para el mantenimiento del software. Lista que será revisada en la fase de revisión funcional al inicio del proyecto:

- Actualización de frameworks (Java, NodeJS, ReactJS, Maven, etc.) a las últimas versiones estables y libres de vulnerabilidades críticas.
- Actualización del SDK de NeuroTecnology para la identificación biométrica por huella.
- Actualización del motor de la base de datos de MongoDB en todos los módulos críticos.
- Análisis de posibles mejoras en el modelo de datos de MongoDB, de cara a mejorar la operación, rendimiento o estabilidad del sistema.
- Actualización de la versión de la imagen de los contenedores docker de los servicios.
- Actualización de los pipelines de despliegue en Jenkins para la integración continua de los diferentes servicios.
- Optimización de modelo de datos del Módulo de Reclamaciones, para mejorar la gestión ante un elevado volumen de solicitudes.
- Analisis para detectar posibles mejoras en el procedimiento de cierre y apertura de la Comisión Nacional, de cara a que sea una operación más veloz.
- Incorporación de analiticas en la web pública, para poder obtener estadísticas de usabilidad.

2.4 Fase de auditoría

Indra contará con sus especialistas en Seguridad, al igual que en 2022, para realizar un análisis de Seguridad estático al código fuente del software de Escrutinio Nacional.

- Revisión completa del código fuente sin necesidad de ejecución, identificando vulnerabilidades en tiempo de desarrollo.
- Detección de errores de seguridad comunes como inyecciones NoSQL, desbordamientos de memoria de la máquina virtual de JAVA, fugas de información sensible, uso inseguro de librerías y dependencias.
- Validación de cumplimiento normativo y buenas prácticas (OWASP Top 10, estándares MinTIC y lineamientos internacionales).
- Generación de informes detallados con clasificación de riesgos, recomendaciones y plan de remediación.

A continuación, se enumeran algunos beneficios de realizar este análisis, cómo práctica habitual, dentro del Ciclo de vida de desarrollo seguro.

- Prevención temprana de vulnerabilidades: Detecta fallos antes de que el software se ejecute, reduciendo costes y tiempos de corrección.
- Mayor robustez y resiliencia del sistema frente a ataques externos y amenazas internas.
- Reducción del riesgo operativo: Evita incidentes que puedan afectar la disponibilidad, integridad o confiabilidad del sistema durante el escrutinio.
- Mejora continua: Permite incorporar prácticas seguras en el ciclo de desarrollo, fortaleciendo la calidad del código.

2.5 Fase de planes de pruebas

Se revisarán los planes de pruebas de 2022, ampliando y corrigiendo lo necesario en función del alcance de cambios de funcionalidad definidos.

- Revisión y actualización de planes existentes: Ajuste de casos de prueba, criterios de aceptación y escenarios críticos según los cambios funcionales y técnicos.
- Definición de nuevos planes de pruebas para módulos actualizados y componentes recién incorporados.

Los planes de pruebas definitivos serán compartidos con la Registraduría.

Los planes de pruebas se implementarán de manera sistemática en cada versión del software que se pretenda cerrar, asegurando que se ejecuten todos los casos definidos en el plan de calidad. El objetivo es alcanzar un nivel óptimo de cobertura que permita validar la funcionalidad, la seguridad y la estabilidad del código, garantizando su conformidad con los requerimientos de la operativa electoral. Este proceso incluirá pruebas funcionales, de integración, rendimiento y regresión, con métricas que certifiquen la trazabilidad y la corrección de incidencias antes de la liberación de cada versión.

Indra dispondrá de la infraestructura de desarrollo propia donde desplegar el SW de escrutinio, con el objetivo de avanzar con la actualización del código fuente, realizando las primeras integraciones funcionales. El entorno de QA y Producción, está fuera del alcance de este proyecto, y será necesario para la ejecución de planes de pruebas, la puesta en producción del Sistema.

2.6 Fase de Mantenimiento Correctivo

En esta fase se aplicarán todas las correcciones derivadas de los procesos de auditoría y pruebas, garantizando que el sistema cumpla con los estándares de calidad, seguridad y rendimiento exigidos por la Registraduría antes de la puesta en producción.

Las actividades incluyen:

- Correcciones de seguridad identificadas en el análisis estático del código (SAST), eliminando vulnerabilidades críticas y ajustando configuraciones inseguras.
- Resolución de errores funcionales detectados en los planes de pruebas funcionales, de integración y en la prueba general del proyecto.
- Correcciones en la integración entre módulos y APIs, asegurando interoperabilidad y consistencia de datos.

- Validación post-corrección mediante ejecución de pruebas para confirmar que los cambios no generan regresiones.
- Test de release en el entorno de producción para asegurar que la versión está lista para su entrega a la RNEC.

2.7 Fase de Entregas

Una vez completadas las fases anteriores, se procederá a la entrega formal del software de Escrutinio Nacional a la Registraduría, asegurando transparencia y trazabilidad.

Esta fase contempla:

- Entrega del software actualizado y validado, incluyendo:
  - Módulos funcionales, con su código fuente.
  - Manuales de usuario y administración.
  - Documentación técnica del sistema, arquitectura y configuración.
  - Informes de auditoría y resultados de pruebas.

3 Mecanismos de control de calidad y validación

Para garantizar la máxima fiabilidad, consistencia y alineación con los objetivos del proyecto, el equipo de Testing & QA de Indra está compuesto por profesionales con amplia y demostrable experiencia en gestión de pruebas, automatización y aseguramiento de la calidad en soluciones electorales.

El equipo participa de forma activa y continua en todo el ciclo de vida del producto, aplicando metodologías y herramientas que aseguran un control integral de la calidad desde la concepción del proyecto hasta su puesta en producción y posterior mantenimiento.

3.1 Metodología de Calidad

3.1.1 Integración temprana del equipo de QA en el ciclo de vida del proyecto

Desde el minuto cero del proyecto, el equipo de Testing & QA trabaja de manera completamente integrada con el equipo de desarrollo y los responsables funcionales. Esta participación temprana permite:

- Conocer de primera mano las especificaciones funcionales y técnicas del producto.
- Identificar riesgos y complejidades desde las primeras etapas.
- Definir criterios de aceptación claros y medibles.
- Establecer la estrategia de pruebas acorde al alcance, reglas de negocio y particularidades del procesamiento de datos.

Durante el despliegue y la puesta en producción, el equipo de pruebas realiza las validaciones funcionales y técnicas finales que garantizan que cada entregable cumple con los requisitos definidos y las expectativas del proyecto.

3.1.2 Definición y ejecución de procedimientos de prueba

Aplicamos un marco metodológico sólido para asegurar la calidad de cada componente y funcionalidad:

- Gestión detallada y trazable de casos de prueba mediante Qase, permitiendo controlar todo el plan de pruebas y cada una de sus iteraciones.
- Gestión de incidencias en Microsoft Planner, donde los defectos se documentan, priorizan y resuelven de forma colaborativa entre todos los equipos implicados.

Este enfoque garantiza un seguimiento preciso del estado de calidad en cada fase, así como la visibilidad del progreso y la evolución del producto.

3.1.3 Validación automatizada de datos procesados

Para asegurar la integridad de los datos y la correcta totalización de resultados:

- Se desarrollan scripts automáticos de validación, diseñados para comprobar consistencia, coherencia y exactitud de la información procesada.
- Estas validaciones permiten detectar desviaciones tempranas y reducen la dependencia de revisiones manuales, incrementando la eficiencia y fiabilidad del sistema.

3.1.4 Validación web multiplataforma y multinavegador

Con el fin de garantizar una experiencia de usuario estable, homogénea y compatible con distintos dispositivos:

- Utilizamos BrowserStack para validar el comportamiento de la web en entornos reales.
- Complementamos este proceso con pruebas automatizadas en Selenium, que validan flujos críticos y elementos visuales en múltiples navegadores y plataformas.

3.1.5 Pruebas de accesibilidad digital con equipo especializado independiente

Un equipo experto, independiente del equipo de pruebas funcionales, realiza evaluaciones exhaustivas de accesibilidad para asegurar que la web pública de resultados sea plenamente funcional para todos los ciudadanos.

Estas pruebas se realizan mediante tecnologías de asistencia como VoiceOver, TalkBack, JAWS y líneas braille, combinadas con herramientas automáticas de análisis.

El proceso se ejecuta en cumplimiento de:

- Las Web Content Accessibility Guidelines (WCAG) en los niveles requeridos.
- La Resolución 1519 de 2020 del Ministerio TIC de Colombia, que regula la accesibilidad digital en servicios del Estado.

3.1.6 Control de calidad tras cada actualización

Cada *release*, despliegue o actualización del sistema incluye:

- Pruebas de regresión manuales y automatizadas.
- Validación de integridad de datos mediante scripts especializados.
- Verificación de nuevas funcionalidades y correcciones aplicadas.
- Validaciones funcionales y técnicas en preproducción.

Estos procesos garantizan la estabilidad del software, la transparencia del cambio y la fiabilidad en todas las entregas.

3.2 Herramientas para la gestión de pruebas

En Indra contamos con una amplia experiencia en herramientas especializadas para la gestión y documentación de pruebas, así como para el tratamiento eficiente de defectos. Para este proyecto proponemos una solución integral basada en Qase, como gestor centralizado de pruebas, y Microsoft Planner, como herramienta de gestión y seguimiento de incidencias.

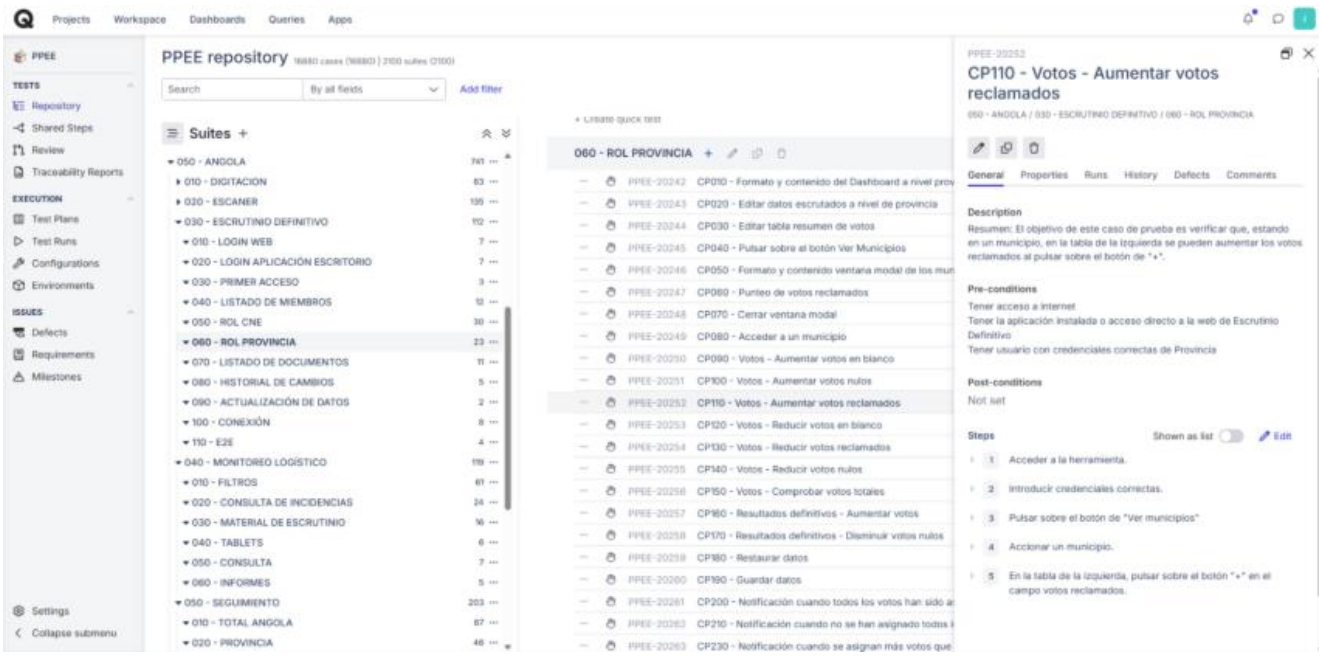
Ambas herramientas se utilizan de forma totalmente integrada, permitiendo una coordinación ágil y un seguimiento end-to-end del ciclo de vida de las pruebas.

3.2.1 Uso de Qase en el proyecto

Qase será la herramienta centralizada para la planificación y ejecución de las pruebas. Su utilización permitirá:

- Documentar y estructurar todos los casos de prueba, agrupándolos en suites y ciclos según los distintos componentes del sistema.
- Mantener trazabilidad directa con los requisitos funcionales, lo que permite identificar claramente qué funcionalidades están cubiertas y en qué estado se encuentran.
- Registrar las ejecuciones de cada iteración, incluyendo evidencias, resultados y bloqueos.
- Disponer de indicadores objetivos de calidad, como porcentaje de pruebas superadas, casos fallidos o módulos con mayor riesgo.

Este enfoque asegura que el avance del plan de pruebas es completamente visible para los equipos implicados y facilita la toma de decisiones durante cada iteración.

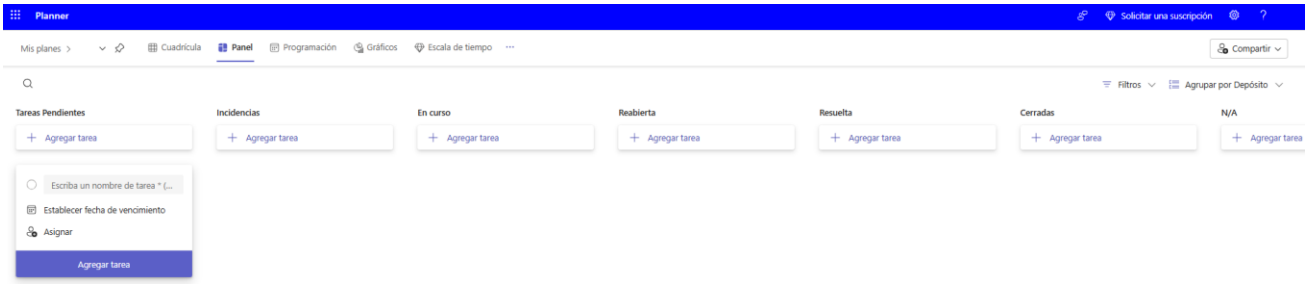


3.2.2 Uso de Microsoft Planner en el proyecto

Microsoft Planner será utilizado como herramienta de registro y seguimiento de incidencias detectadas durante la ejecución de las pruebas. Su incorporación permite:

- Organizar las incidencias en un flujo de trabajo claro, desde su creación hasta su cierre.
- Asignar, priorizar y revisar correcciones de forma coordinada entre QA, desarrollo y responsables funcionales.
- Asegurar la trazabilidad del defecto, relacionándolo directamente con el caso de prueba que lo originó.
- Disponer de un tablero visual sencillo, que facilita la identificación del estado de cada incidencia y el volumen de trabajo pendiente en cada equipo.

Microsoft Planner proporciona una gestión ágil y accesible para todos los equipos involucrados, reduciendo tiempos de resolución y evitando duplicidades.



3.2.3 Integración operativa entre Qase y Planner

El uso combinado de ambas herramientas aporta coherencia y continuidad al proceso de pruebas:

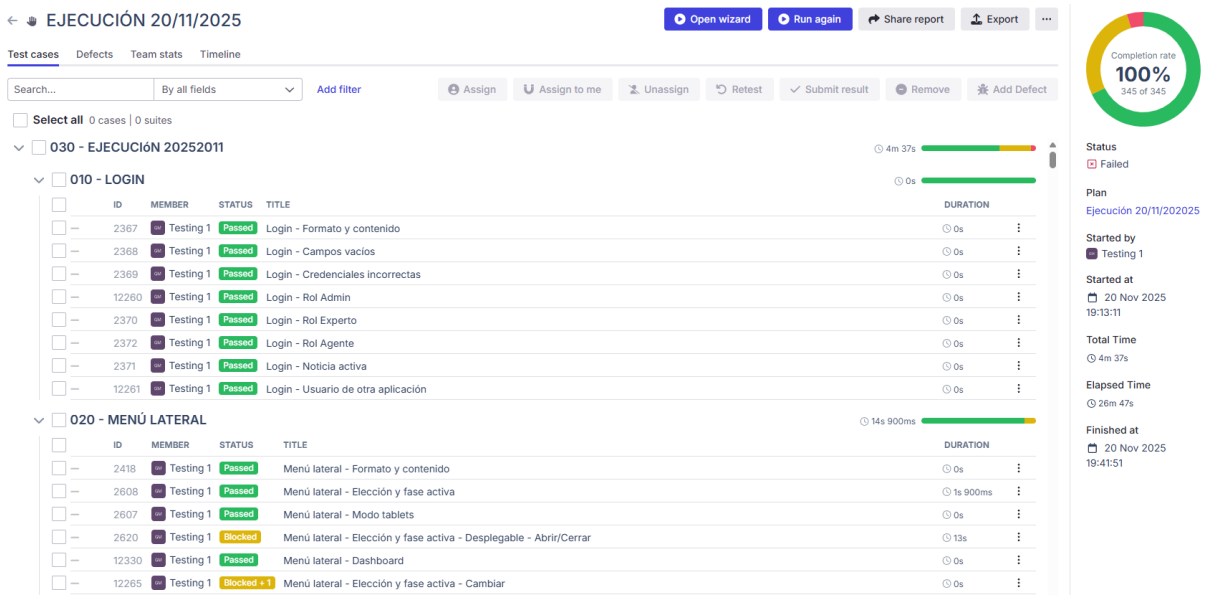
- Los casos de prueba se ejecutan en Qase.
- Cuando se detecta un defecto, se registra en Planner, indicando su severidad, prioridad, descripción y referencia al caso que lo generó.
- El equipo de desarrollo actúa sobre la incidencia, actualizando el estado según corresponda.
- Cuando la corrección está disponible, el equipo de testing verifica la corrección del caso y se notifica en Qase, cerrando el ciclo completo.

Esta integración evita duplicidad de información, facilita la comunicación entre equipos y garantiza que cada defecto queda completamente trazado desde su detección hasta su resolución.

3.2.4 Beneficios para la gestión del proyecto

El uso coordinado de Qase y Planner aporta mejoras concretas al desarrollo del proyecto:

- Trazabilidad completa entre requisitos, casos de prueba e incidencias.
- Visibilidad continua del estado de calidad, permitiendo anticipar riesgos y ajustar prioridades.
- Reducción de tiempos de comunicación entre QA y desarrollo, gracias a flujos claros y centralizados.
- Control sobre la cobertura de pruebas, tanto por módulos como por entregables.
- Soporte para auditorías o revisiones externas, al contar con un histórico completo y estructurado.



En conjunto, estas herramientas permiten gestionar el proceso de pruebas de forma ordenada, transparente y eficiente, asegurando que cada entrega del sistema se valida con criterios homogéneos y plenamente trazables.

3.3 Herramientas de automatización

Para optimizar la ejecución de pruebas repetitivas, minimizar errores humanos y garantizar una validación constante en diferentes versiones del software, el proyecto contará con un conjunto de herramientas de automatización que cubren tanto pruebas web como en dispositivos móviles, así como validaciones de datos y compatibilidad entre plataformas.

Estas herramientas permiten escalar las pruebas, acelerar los ciclos de validación, mejorar la calidad global del producto y optimizar la priorización y la gestión de recursos.

A continuación, se detalla el papel de cada una dentro del proyecto.

3.3.1 Selenium

Selenium se utilizará como herramienta principal para la automatización de pruebas funcionales en aplicaciones web. Su uso permitirá:

- Automatizar flujos críticos del sistema (consulta, navegación, interacción con formularios, validación de datos).
- Validar interfaces en diferentes navegadores y versiones.
- Ejecutar pruebas de forma repetible y estable en cada release o despliegue.
- Reducir el tiempo necesario para validar regresiones en la parte web.

Las pruebas automatizadas se integrarán en los ciclos de validación, garantizando que las funcionalidades clave se verifican de manera consistente en todas las iteraciones.

3.3.2 Appium

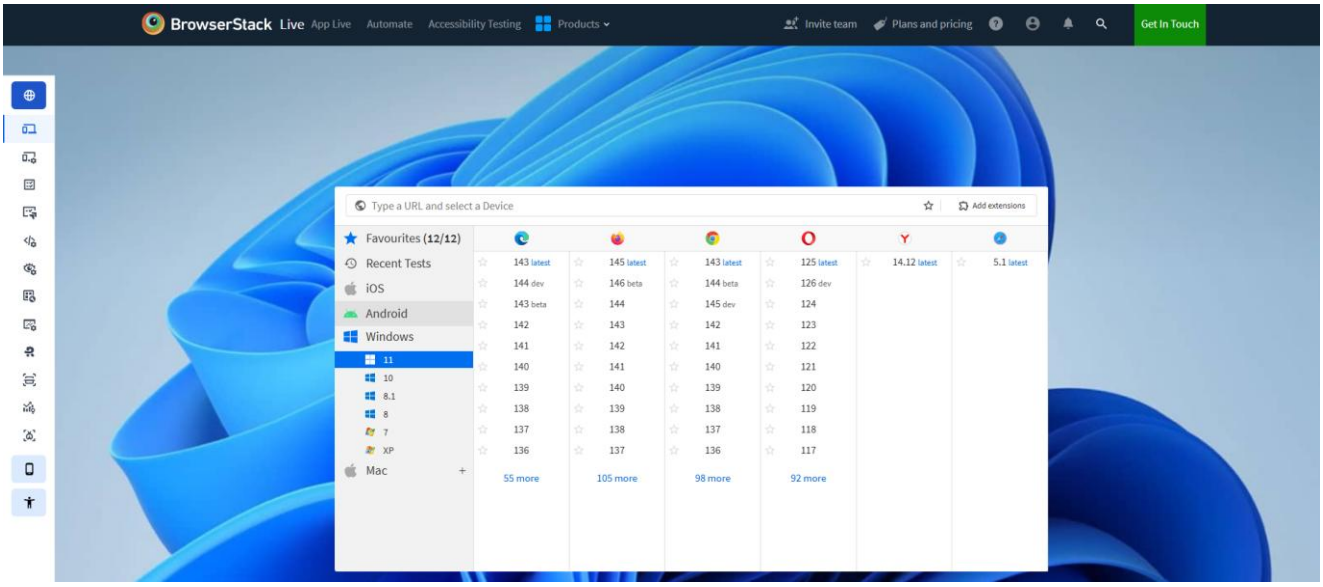
Appium permitirá automatizar las pruebas en dispositivos móviles, tanto nativas como web, aportando:

- Simulación de interacciones reales del usuario (toques, gestos, desplazamientos).
- Validación en múltiples modelos de dispositivos, resoluciones y sistemas operativos.
- Ejecución de flujos repetitivos sin depender de validaciones manuales.
- Detección rápida de fallos relacionados con interfaz, rendimiento básico o compatibilidad.

Appium y Selenium se integrará también con plataformas de laboratorio de dispositivos, como las incluidas en BrowserStack, para ampliar aún más el factor de forma de los dispositivos y los entornos de prueba.

3.3.3 BrowserStack

BrowserStack es una plataforma que permite ejecutar pruebas —tanto manuales como automatizadas— en un amplio catálogo de navegadores con sus diferentes versiones, diferentes sistemas operativos y dispositivos reales.



Su integración con Selenium y Appium permite extender las automatizaciones ya desarrolladas a entornos reales sin necesidad de disponer físicamente de esos dispositivos.

Dentro del proyecto, BrowserStack aporta:

- Ejecución de pruebas en múltiples navegadores y versiones, garantizando compatibilidad y estabilidad.
- Validación en dispositivos móviles reales y sistemas operativos diversos.
- Reproducción de incidencias detectadas en entornos reales para su análisis y resolución.
- Reducción de costes y tiempos asociados a la gestión de un laboratorio físico de dispositivos.

Su uso asegura que la solución funcione de manera consistente en todos los escenarios previstos de uso final.

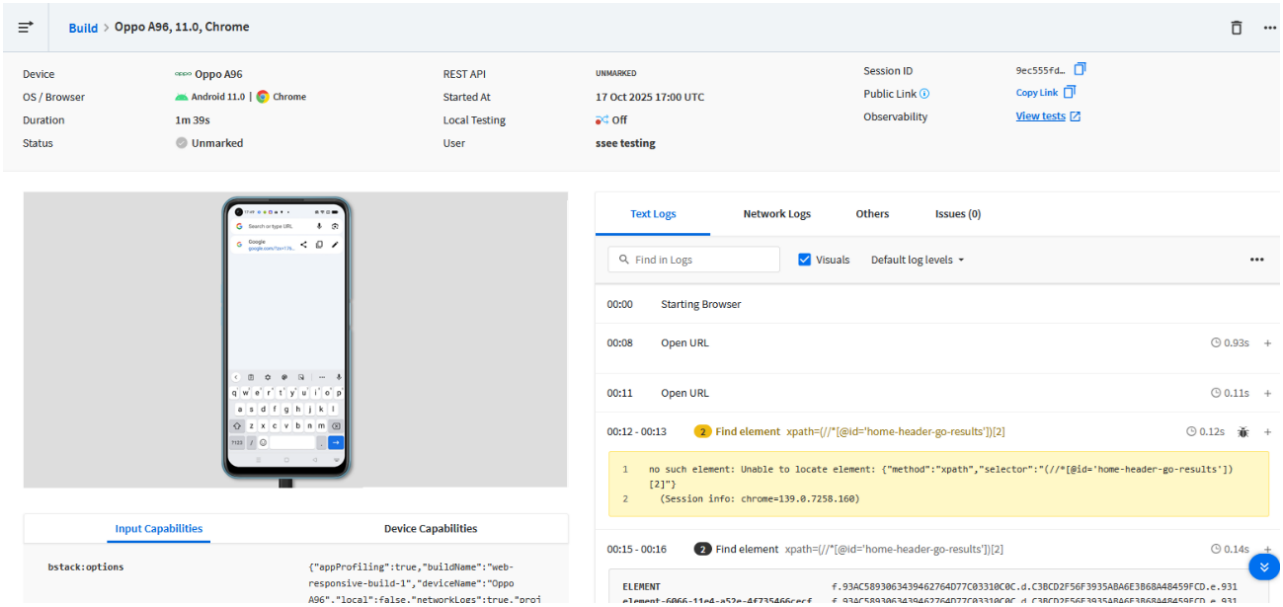
3.3.4 Uso conjunto de las herramientas en el proyecto

Las herramientas descritas no se utilizan de forma aislada, sino integradas dentro del proceso de calidad del proyecto:

- Selenium automatiza las pruebas funcionales web.
- Appium automatiza las pruebas en dispositivos móviles.
- BrowserStack ejecuta los scripts en múltiples navegadores y dispositivos reales.

Este enfoque integrado garantiza:

- Cobertura completa de validación: web, móvil y datos.
- Reducción de tiempos en regresiones y validaciones repetitivas.
- Ejecución homogénea del mismo conjunto de pruebas en diferentes plataformas.
- Detectar errores antes, con mayor precisión y con menor coste operativo.



3.3.5 Beneficios directos para la gestión del proyecto

La combinación de Selenium, Appium y BrowserStack ofrece ventajas claras para la ejecución y seguimiento del proyecto:

- Mayor fiabilidad del producto al validar de forma automatizada funcionalidades críticas en cada entrega.
- Reducción de tiempos de validación, permitiendo ciclos más rápidos y mayor capacidad de reacción ante incidencias.
- Cobertura ampliada, asegurando que cada versión se prueba en múltiples dispositivos y configuraciones sin esfuerzo adicional.
- Trazabilidad completa de las pruebas automatizadas dentro del proceso general de QA.
- Menor riesgo operativo, al disminuir dependencias de recursos manuales y validar el comportamiento del sistema en escenarios reales.

En conjunto, esta estrategia de automatización contribuye de manera directa a mejorar la estabilidad del sistema, a reducir la aparición de incidencias en producción y a garantizar una validación consistente y escalable a lo largo del proyecto.

3.4 Herramientas de accesibilidad

Para garantizar el cumplimiento de los requisitos de accesibilidad establecidos por la Administración —incluyendo ONTI, OAW y la Resolución 1519 de 2020 del MinTIC— nuestro equipo especializado e independiente utiliza un conjunto diverso de herramientas adaptadas a cada tipo de plataforma evaluada. Esto permite asegurar que todas las evaluaciones consideren distintos perfiles de usuario, necesidades específicas y escenarios de uso reales.

3.4.1 Herramientas comunes

- Colour Contrast Analyser: Evalúa la relación de contraste entre texto y fondo, asegurando la legibilidad según los estándares internacionales.
- Línea braille Focus: Facilita la verificación de contenido para usuarios con discapacidad visual, permitiendo revisar la correcta representación en dispositivos braille.

3.4.2 Herramientas para web

Se emplean lectores de pantalla, ampliadores de contenido y sistemas de navegación asistida que permiten simular la experiencia de distintos usuarios:

- JAWS (Job Access With Speech): Permite evaluar la accesibilidad mediante lectura auditiva de contenidos web, navegación mediante teclado y compatibilidad con etiquetas ARIA.
- ZoomText Magnifier: Evalúa la experiencia de usuarios con visión reducida mediante ampliación de texto e imágenes, así como ajustes de color y enfoque.

3.4.3 Herramientas para dispositivos móviles

Se aprovechan las funciones nativas de accesibilidad de los sistemas operativos móviles para asegurar una experiencia inclusiva:

- Funciones de accesibilidad de iOS: Incluyen VoiceOver, lupa, subtítulos y ajustes de contraste y tamaño de texto, entre otros.
- Suite de accesibilidad de Android: Comprende TalkBack, ampliación de pantalla, controles de gestos, subtítulos y otras funcionalidades que permiten evaluar distintos perfiles de usuario.

El uso combinado de estas herramientas permite realizar evaluaciones completas y precisas, garantizando que las aplicaciones y plataformas cumplan con los estándares de accesibilidad vigentes y ofrezcan una experiencia inclusiva a todos los usuarios, independientemente de sus capacidades.

4 Metodologías de desarrollo

Indra establecerá la metodología de gestión y de desarrollo, que se adapta a las características de este proyecto, cumpliendo con el estándar de calidad CMMI. La empresa cuenta con la certificación en el nivel 5 de grado de madurez CCMI, donde determinados procesos deberán estar definidos, controlados, medidos y documentados, entre los que destacan los siguientes en el contexto del proyecto:

- Análisis Causal y Resolución: identificar las causas de los resultados derivados de las pruebas realizadas sobre el sistema, y actuar para mejorar el rendimiento de proceso.
- Gestión de la configuración: establecer y mantener la integridad de las aplicaciones llevando un control de la configuración de las mismas, así como control de versiones y despliegue en la plataforma.
- Gestión Integrada del Proyecto: establecer y gestionar el proyecto y la involucración del cliente en la gestión del proyecto, permitiendo el acceso a las herramientas
- Definición del proceso de la organización: establecer metodologías de trabajo, reglas y normas de trabajo que serán cumplidas por las diferentes áreas.
- Formación en la organización: desarrollar las habilidades y los conocimientos de las personas para que puedan desempeñar sus roles eficaz y eficientemente. Fomentar la transmisión de conocimiento, para que las funciones de un perfil puedan ser sustituidas en caso de necesidad.
- Integración del producto: ensamblar las aplicaciones a través de los diferentes módulos, asegurando que una vez integrado cumple con todas las funcionalidades, para lo que se ejecutarán pruebas de integración en cada despliegue de versión.
- Monitorización y control del proyecto: proporcionar herramientas para el monitoreo y administración de los diferentes módulos del sistema.
- Planificación del proyecto: establecer y mantener planes que definan las actividades del proyecto.
- Gestión cuantitativa del proyecto: gestionar cuantitativamente el proyecto para alcanzar los objetivos establecidos de calidad y de rendimiento de proceso en el proyecto, para lo que se realizará un plan de pruebas que contemple todas las casuísticas de sistema y de aplicación posibles.
- Desarrollo de Requisitos: analizar y establecer de manera coordinada con el cliente los requisitos del servicio.
- Gestión de requisitos: gestionar los requisitos de los productos y los componentes de producto del proyecto, y asegurar la alineación entre esos requisitos, y los planes y los productos de trabajo del proyecto.
- Gestión de Riesgos: identificar problemas potenciales antes de que ocurran, analizando los retrasos y los cuellos de botella que se están generando en el proyecto.
- Solución técnica: diseñar e implementar los sistemas en función de los requisitos.
- Validación y Verificación: demostrar que un producto o componente de producto cumple con su uso y funcionalidades previstas, con la realización del plan de pruebas, entre las que se incluye un conjunto de pruebas funcionales del 100%.

Las buenas prácticas del modelo CMMI para el desarrollo de Software definen que se tengan en cuenta las siguientes actividades.

- Implantar los procesos necesarios para la gestión del proyecto, que permita controlar todos los integrantes del mismo.
- Desarrollar el código cumpliendo con el ciclo de vida de desarrollo y utilizando patrones de diseño.

- Utilizar herramientas para la trazabilidad, auditoría y seguridad del sistema, que garanticen la integridad de la información.
- Documentar el software a lo largo del proyecto, con análisis de requerimientos, análisis funcional, diagramas de componentes, entre otros.
- Llevar un control de la liberación de versiones, con un sistema de despliegue. Utilizar entornos separados de desarrollo, QA o preproducción, y producción.



4.1 Metodología ágil

El desarrollo de software enfocado a los procesos electorales tiene ciertas características diferentes a otro tipo de mercados. Los periodos de implementación son cortos, y las especificaciones muy variables, por lo que es importante que los sistemas sean adaptables y configurables. La entrega de las aplicaciones no se puede retrasar, y se debe cumplir con la planificación fijada.

Indra, con la experiencia en la ejecución de este tipo de proyectos, propone la utilización de metodologías ágiles. Este tipo de metodologías permitirán gestionar el proyecto más dinámica y eficazmente, maximizar la producción del equipo y responder más rápidamente a los ajustes necesarios a lo largo del desarrollo.

Se seguirá la evolución del proyecto día a día, en comunicación directa y continua con el cliente.

De las metodologías ágiles más populares, se seguirá la filosofía de Scrum, con su carácter iterativo basado en la entrega progresiva de productos testeables. Además, el día a día del proyecto se apoyará en a la metodología enfocada al manejo de tableros Kanban, que podrá ser utilizada tanto en los

procesos de gestión como en el resto de las tareas del proyecto sin importar su tipología, permitiendo así la integración de los métodos de trabajo entre todo el equipo del proyecto

Otras de las ventajas de esta metodología:

- La utilización de tableros permite aumentar la eficiencia del trabajo en contextos donde la mayor parte del trabajo es operativa. Muy veloz para tareas con poca descripción, suficiente para que un desarrollador se pueda hacer cargo de su implementación
- Permite visualizar todo lo que está ocurriendo en un determinado momento. Cada tarea y su estado de avance se ve en el contexto de todo el proyecto.
- Disponibilidad de un "backlog" de tareas, categorizadas, priorizadas y administradas, que almacenará las tareas pendientes del proyecto.
- Gestionar la capacidad del trabajo en curso (Work-In-Progress), visualizando los posibles cuellos de botella en alguna área, facilitando la toma de decisiones.

4.2 Ciclo de vida de desarrollo de software seguro

Implementamos metodologías de ciclo de vida de desarrollo de software seguro, que proporcionan un marco integrado de mejora de la seguridad utilizando políticas de seguridad, estándares de calidad y los siguientes procedimientos específicos:

- Formación y concienciación. Todo el personal del proyecto debe tener conocimientos suficientes para abordar sus responsabilidades desde un punto de vista más completo que incluya la perspectiva de la seguridad en los servicios a implementar.
- Análisis de Requerimientos de Seguridad. Se incluyen los requerimientos de seguridad en una fase temprana para evitar el impacto tan elevado que supone la modificación de los componentes ya desarrollados, al tener que corregir vulnerabilidades detectadas en fases posteriores.
- Patrones de Diseño y Patrones de Arquitectura. La utilización de los patrones de diseño y de arquitectura más apropiados garantiza que se utilicen soluciones comunes ante problemas comunes, evitando algunas implementaciones realmente complejas o enrevesadas.
- Herramientas de Análisis Estático de Código. Se utilizan herramientas que comprueban el código fuente a fondo en busca de posibles problemas de seguridad en una fase temprana.
- Integración Continua. Utilizamos herramientas de integración continua que automaticen los empaquetados y despliegues en los diferentes entornos, así como la ejecución de pruebas unitarias, de integración y regresión.
- Test de Intrusión. Las herramientas automáticas de intrusión son imprescindibles como táctica de defensa en profundidad para detectar fallos de seguridad que no hayan sido detectados en fases anteriores.
- Identificación y Autenticación. Los módulos desarrollados siempre deben garantizar que el usuario que trata de acceder al sistema es identificado y además es quien dice ser presentando sus credenciales a través de diferentes mecanismos.
- Autorización y Control de Acceso. El sistema debe permitir a sus usuarios realizar únicamente las acciones para las que están autorizados. También debe garantizar que sus recursos están debidamente protegidos.
- Trazabilidad y Auditoría. Todas las acciones que se produzcan en los sistemas deben quedar debidamente registradas, para garantizar que es posible identificar íntegramente al usuario o sistema que las origina.

- Integración con Pruebas de Seguridad. Se incluyen pruebas de seguridad en las herramientas de integración continua y en los planes de pruebas, para verificar que se cumplen correctamente todos los requerimientos de seguridad definidos.
- Empleo de Técnicas de Codificación Segura:
- Manejo de entradas de datos. Actividades que aceptan los datos de entrada sin confiar en ellos y comprueban la validez de los mismos, limitándolos únicamente a los valores que son aceptables.
- Desbordamientos de buffers. Comportamiento que ocurre cuando un programa escribe los datos fuera de los límites de la memoria asignada para ello.
- Errores de truncado y conversión. Manejo de enteros con y sin signo y de diferente tamaño.
- Manejo de errores y excepciones. Verificación de todos los posibles errores y excepciones que garantice que el sistema se queda en un estado seguro tras el fallo.
- Uso de privilegios de los programas. Control de las posibilidades que tienen los usuarios legítimos de realizar acciones no autorizadas a través de debilidades en los recursos.

4.3 Control de versiones

El equipo de desarrollo de aplicaciones utilizará un control de versiones para la fase de implementación del servicio y las mejoras que se vayan introduciendo en el mantenimiento del mismo.

Como herramienta de control de versiones seleccionada para el proyecto, se utiliza GIT.

Para ello:

- Cada aplicación contará con una línea de versionado de Preproducción (QA).
- La línea de QA será bifurcada en ramas de desarrollo por funcionalidad independiente. Cuando la funcionalidad esté probada y sea estable, podrá integrarse con la línea base de QA.
- De esa manera los desarrolladores no se verán afectados por el resto de las implementaciones, evitando la propagación de errores.
- La herramienta de control de versiones permitirá la trazabilidad de cualquier cambio que se haga en las aplicaciones, así como un registro histórico de las versiones que se han ido desarrollando.
- Se realizará control de versiones de los templates de despliegue de cada aplicación en el cluster de Kubernetes y de los archivos de configuración en cada entorno.

4.4 Seguridad del código

Con el objetivo de minimizar los posibles errores de las aplicaciones, mejorar la calidad del código escrito, reducir las posibles vulnerabilidades de seguridad o malas prácticas, Indra usará la herramienta de evaluación de código estático (sin ejecución) Sonarqube, en las aplicaciones que conforman el proyecto.

Esta herramienta se enlazará con los repositorios del control de versiones, para evaluar el código cada vez que se haga un despliegue en QA, a través de las reglas configuradas a nivel de calidad, mantenibilidad, confiabilidad y seguridad, según los estándares de la OWASP (Open Web Application Security Project) o CWE (Common Weakness Enumeration), tanto para las APIs como para las aplicaciones web. Este análisis de código estático se realizará en la herramienta de integración continua del proyecto (Jenkins), previo a la actualización de una nueva versión de las aplicaciones en los contenedores (Docker) utilizados.

Además, la herramienta de integración analizará y evaluará políticas de seguridad en las imágenes de los contenedores antes de desplegarlo en el orquestador de recursos de la infraestructura (kubernetes).

Algunas políticas que se comprobarán son:

- Análisis de vulnerabilidades de los paquetes utilizados en la imagen de Docker.
- Puertos expuestos en el contenedor.
- Credenciales en texto plano en el contenedor.

4.5 Integración continua

Se utilizará una herramienta como método de integración continua para evaluar la compatibilidad de los diferentes desarrollos.

La herramienta elegida como servidor de automatización de despliegues es Jenkins.

Esta herramienta aporta las siguientes ventajas a la ingeniería del software del proyecto:

- Automatización de los despliegues de software en los diferentes entornos, evitando posibles errores de la gestión manual y garantizando que las operaciones de despliegue siempre sean las mismas.
- Control de usuario y configuración de roles, filtrando el acceso a los despliegues en los diferentes entornos.
- Auditoría de los despliegues realizados, visualizando el histórico de trabajos realizados y los commits del git que contiene cada despliegue.
- Ejecución de pruebas unitarias automáticas al finalizar el despliegue, para evaluar el estado de la aplicación.
- Integración con las tecnologías elegidas para el proyecto (javascript, nodejs, java, springboot).

Entornos y Namespaces

- Desarrollo. Donde trabajan los desarrolladores del equipo para avanzar con la implementación de las funcionalidades. Los desarrolladores dispondrán de su entorno en local, además de un namespace en el cluster de Kubernetes donde probar las funcionalidades y hacer las integraciones de código.
- QA. Entorno donde se integran las aplicaciones y donde el equipo de testing ejecuta los planes de pruebas al sistema completo.
- Producción. Entorno de ejecución real donde accede la Comisión Nacional a utilizar el Sistema de Escrutinio.
- Capacitación. Entorno paralelo y estable utilizado para alguna capacitación a usuarios del proyecto. La versión desplegada en este entorno debe contemplar la funcionalidad necesaria para la capacitación y debe ser estable para no afectar a la disponibilidad durante horarios formativos.

El entorno de desarrollo se ubicará en infraestructura propia de Indra, mientras que el resto de entornos, se ubicarán en la infraestructura productiva del CNE, fuera del alcance de este contrato.

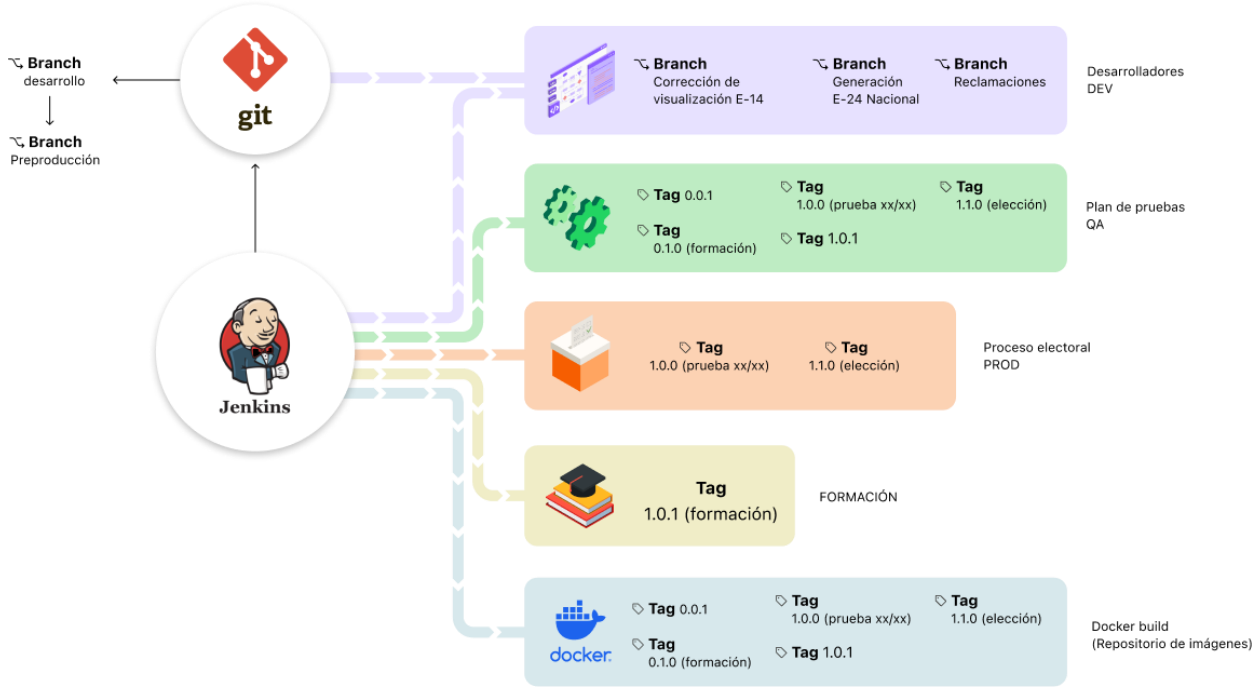
Despliegues

Mediante la herramienta de Jenkins, se definirá un despliegue para cada servicio del sistema.

- Despliegues en el entorno de Desarrollo:

- Se realizarán integraciones sobre las ramas de desarrollo asociadas a cada nueva funcionalidad o correcciones.
- Cuando se lanza el despliegue de desarrollo, Jenkins crea un nuevo trabajo y descarga las fuentes de la rama del GIT, de la rama de desarrollo.
- Compila y empaqueta las fuentes, según la tecnología utilizada.
- Genera la imagen del docker y la sube al repositorio de imágenes. La numeración de la versión se incrementa un 0.0.1 sobre la última versión desplegada en todos los entornos.
- Jenkins despliega en el clúster de kubernetes, en el namespace de DEV, la imagen de docker generada, con el ConfigMap asociados al entorno de desarrollo.
- Despliegues en el entorno de QA:
- Se realizará un mergeo de la rama de desarrollo a la rama de preproducción
- Cuando se lanza el despliegue de QA, Jenkins crea un nuevo trabajo y descarga las fuentes de la rama del GIT, de la rama de preproducción.
- Esta imagen es etiquetada con un número de versión, incrementando en 0.1.0 al valor actual y colocando el tag de “stable”.
- Jenkins despliega en el clúster de kubernetes, en el namespace de QA, la imagen de docker generada, con el ConfigMap asociado al entorno de QA.
- El equipo de testing se encarga de certificar el número de versión ejecutando el plan de pruebas correspondiente en el entorno de QA. Una vez certificada la idoneidad de la versión, se puede lanzar el despliegue en Producción.
- Despliegues en el entorno de Producción:
- No se realiza ninguna compilación de código desde las ramas del GIT.
- Se recupera del repositorio de Docker la última imagen disponible con tag “stable”.
- Esta imagen es etiquetada con un número de versión, incrementando en 1.0.0 al valor actual y colocando el tag de “latest”.
- Jenkins despliega en el clúster de kubernetes, en el namespace de PROD, la imagen de docker generada, con el ConfigMap asociado al entorno de producción.
- Existe la garantía de que se despliega en producción la misma versión de Docker de QA.
- En caso de que ocurra algún error, el sistema recupera la versión anterior en Producción y la despliega de nuevo.

También se puede disponer de algún entorno de formación, por si hubiera que capacitar a la CNE en la utilización del Sistema.


Branch
desarrollo
Branch
Preproducción
git
Jenkins
Branch
Corrección de visualización E-14
Branch
Generación E-24 Nacional
Branch
Reclamaciones
Desarrolladores DEV
Tag 0.0.1
Tag 0.1.0 (formación)
Tag 1.0.0 (prueba xx/xx)
Tag 1.0.1
Tag 1.1.0 (elección)
Plan de pruebas QA
Tag 1.0.0 (prueba xx/xx)
Tag 1.1.0 (elección)
Proceso electoral PROD
Tag 1.0.1 (formación)
FORMACIÓN
docker
Tag 0.0.1
Tag 0.1.0 (formación)
Tag 1.0.0 (prueba xx/xx)
Tag 1.0.1
Tag 1.1.0 (elección)
Docker build (Repositorio de imágenes)

REGISTRADURÍA
NACIONAL DEL ESTADO CIVIL

INDRA COLOMBIA S.A.S.