PLAN DE PRUEBAS
YO CUIDO LO PÚBLICO ADMINISTRADOR
SOLUCIONES MÓVILES 4
PROYECTO FÁBRICA DE SOFTWARE GRUPO 2

Soluciones y Servicios Tecnológicos
Dirección de Gobierno en línea

Bogotá, D.C, abril de 2014

FORMATO PRELIMINAR AL DOCUMENTO

Título:	**PLAN DE PRUEBAS**				
Fecha elaboración aaaa-mm-dd:	2014-02-07				
Sumario:	El plan de pruebas describe de manera general y detallada el propósito y funcionalidad del proceso de pruebas de la aplicación Yo Cuido Lo Público Administrador del proyecto Soluciones Móviles4.				
Palabras Claves:	Plan de pruebas, pruebas funcionales, pruebas no funcionales.				
Formato:	DOC		Lenguaje:		Español
Dependencia:	Ministerio de Tecnologías de la Información y las Comunicaciones: Dirección de Gobierno en línea – Soluciones y Servicios Tecnológicos				
Código:	GLFS2-SM4-PPR	Versión:	2.0	Estado:	Aprobado
Categoría:					
Autor (es):	Cristina Cortes Albadan Líder Técnico UT Software Works		Firmas:		
Revisó:	Mónica Monroy Consultor Procedimientos y Herramientas de Interventoría Consorcio S&M Jorge Santiago Moreno Dirección Gobierno en línea Luisa Fernanda Medina Dirección Gobierno en línea Fernando Segura Asesor Secretaría de Transparencia				
Aprobó:	Luis Felipe Galeano Arquitecto IT Consorcio S&M Rafael Londoño Dirección de Gobierno en línea				
Información Adicional:	No Aplica				
Ubicación:	El archivo magnético asociado al documento está localizado en el repositorio 24 – SOLUCIONES MOVILES 4 en la siguiente ruta: 03. Fase de Ejecución / 02. Diseño / 03. Plan de Pruebas / 01. Plan de Pruebas				

CONTROL DE CAMBIOS

VERSIÓN	FECHA	No. SOLICITUD	RESPONSABLE	DESCRIPCIÓN
1.0	2014-02-07	No aplica	UT Software Works	Elaboración del documento
1.1	2014-02-26	No aplica	UT Software Works	Ajustes solicitados por interventoría a la sección 4.2.2 Pruebas No Funcionales.
1.2	2014-04-14	No aplica	UT Software Works	Ajustes solicitados por interventoría GEL y Entidad
2.0	2014-04-22	No aplica	UT Software Works	Aprobación del documento

TABLA DE CONTENIDO

LISTA DE TABLAS

DERECHOS DE AUTOR

CRÉDITOS

En un trabajo conjunto entre los consultores de la Dirección de Gobierno en línea, las firmas Secretaría de Transparencia, Consorcio S&M y la UT Software Works, se ha generado el presente documento siguiendo los estándares establecidos en el Sistema de Gestión de Calidad de la Dirección de Gobierno en línea, para el proyecto **IMPLEMENTACIÓN DE SOLUCIONES TECNOLÓGICAS BAJO EL MODELO DE FÁBRICA DE SOFTWARE PARA LAS INICIATIVAS DEL PLAN VIVE DIGITAL A CARGO DEL PROGRAMA AGENDA DE CONECTIVIDAD Y LA EVOLUCION DE LAS SOLUCIONES QUE SOPORTAN LA ESTRATEGIA DE GOBIERNO EN LÍNEA GRUPO 2.**

Este documento fue revisado y aprobado por los consultores y profesionales de la Dirección de Gobierno en línea, previa validación de la empresa interventora del contrato Consorcio S&M.

1. AUDIENCIA

Este documento está dirigido a los integrantes de los equipos de la Dirección de Gobierno en línea, el Consorcio S&M , Secretaria de Transparencia y la Unión Temporal UT Software Works que participan en el proyecto. Este documento es aplicable a la solución Yo Cuido Lo Público Web del proyecto soluciones móviles 4 , el cual debe ser conocido por los miembros de los equipos del proyecto: **IMPLEMENTACIÓN DE SOLUCIONES TECNOLÓGICAS BAJO EL MODELO DE FÁBRICA DE SOFTWARE PARA LAS INICIATIVAS DEL PLAN VIVE DIGITAL A CARGO DEL PROGRAMA AGENDA DE CONECTIVIDAD Y LA EVOLUCION DE LAS SOLUCIONES QUE SOPORTAN LA ESTRATEGIA DE GOBIERNO EN LÍNEA GRUPO 2.**

2. INTRODUCCIÓN

El presente documento presenta el plan de pruebas para satisfacer los requisitos de calidad de la solución asegurando la satisfacción de la Dirección de Gobierno en línea.

El propósito del Plan de pruebas es proveer un artefacto central que gobierne la planeación y control del esfuerzo de pruebas. Este define el enfoque general que será empleado para probar el software y para evaluar los resultados de esas pruebas y es el plan de más alto nivel que será usado por los administradores para guiar y dirigir el trabajo de pruebas en detalle.

3. ALCANCE

Este documento detalla las actividades necesarias para ejecutar los tipos de pruebas que se realizarán para el proyecto Soluciones Móviles 4 solución Yo Cuido Lo Público Administrador Web desarrollada por la UT Software Works. El alcance de las pruebas está enfocado en probar todas las historias de usuario y requerimientos no funcionales desarrollados para la solución.

4. PLAN DE PRUEBAS DETALLADO

4.1 METODOLOGÍA DE PRUEBAS

A continuación, se presenta la metodología de pruebas por cada uno de los ambientes de prueba que se tienen definidos:

- **Ambiente local (ambiente de desarrollo)**

En este ambiente se realizan las pruebas unitarias por parte de los desarrolladores, las pruebas se realizarán usando la herramienta Visual Studio .Net 2012.

Estas pruebas deben ser realizadas antes de liberar versiones en el ambiente de pruebas interno (pruebas UTSW) para pruebas internas de la UTSW.

- **Ambiente pruebas interno (pruebas UTSW)**

En este ambiente se realizan las pruebas internas por parte de las personas asignadas a las pruebas de la solución por parte de la UTSW.

Para ejecutar las pruebas en este ambiente se deben diseñar los casos de prueba que se requieran para cada una de las historias de usuario que se van a probar en el sprint. El diseño de casos de prueba se realizará teniendo en cuenta la plantilla GELFS2-GB-DPR-CasosDePrueba que se encuentra en el repositorio 00-FABRICA DE SOFTWARE en la ruta MODULO DOCUMENTOS / Administración / 2. Planeación / Plan de Trabajo / Anexos / Formatos/Plantillas / Otros.

Los ciclos de prueba previstos para ejecutarse en este ambiente son:

- Sprint 1 al antepenúltimo
  - Se ejecutan todos los casos de prueba que se tengan definidos para las historias de usuario realizadas en el sprint.
- Sprint revisión final de calidad
  - Se ejecutan todos los casos de prueba que se tengan definidos para las historias de usuario realizadas en el sprint.

- **Ambiente de pruebas preproducción**

Cuando la solución ha terminado el último sprint de construcción se inician las pruebas en el ambiente de preproducción, las cuales se realizan en dos fases, la fase 1 para que la UTSW realice una revisión final de calidad antes de liberar la versión final a pruebas por parte de Interventoría/Entidad/GEL y la fase 2 donde el la UTSW verifica que la versión se encuentra correctamente instalada y notifica a Interventoría/Entidad/GEL que se puede dar inicio a las pruebas que ellos realizan, teniendo en cuenta lo que se indica a continuación para cada una de las fases.

  - **Fase 1**

    En esta fase se realizan pruebas por parte de la UTSW.

    Si el ambiente de Preproducción es provisto por la UTSW se realizará la actividad de Revisión Final de Calidad en el ambiente de preproducción, para la ejecución de pruebas se realizarán dos ciclos de prueba para ejecutar las pruebas Funcionales y No funcionales definidas en el presente documento. La versión a probar será la que se liberará para pruebas por parte de Interventoría/Entidad/GEL en el Sprint de Preproducción.

  - **Fase 2**

    En esta fase se realizan pruebas por parte de la Interventoría/Entidad/GEL una vez la UTSW ha verificado que la versión se encuentra correctamente instalada.

    Si el ambiente de preproducción es provisto por la UTSW entonces la UTSW:

    - Verifica que la versión corresponde con la versión que debe probar la Interventoría/Entidad/GEL.
    - Notifica por medio de comunicado a Interventoría/Entidad/GEL que la versión se encuentra instalada en Preproducción para pruebas por parte de Interventoría/Entidad/GEL.

- **Ambiente de Producción:**

El equipo de pruebas de la UTSW realiza una verificación rápida para asegurar que la versión se encuentra bien instalada antes de iniciar las pruebas que realizará en este ambiente.

Las pruebas en este ambiente por parte de la UTSW se iniciarán una vez se instale la versión en el ambiente de Producción.

4.2 TIPOS DE PRUEBAS

4.2.1 PRUEBAS FUNCIONALES

Pretende validar los requisitos funcionales, incluyendo la navegación dentro del sistema, entrada de datos, procesamiento y obtención de resultados.

- **Pruebas unitarias:** Se verifica la funcionalidad de cada módulo.
- **Pruebas de funcionalidad:** Se verifica que la funcionalidad de la aplicación satisface los requerimientos funcionales solicitados.
- **Pruebas de interfaz de usuario:** Se verifica que el componente de la interfaz gráfica de usuario sea consistente (elementos de navegación, presentación de iconos, nombre de los elementos, ortografía de los elementos, entre otros.).
- **Pruebas de ciclo de negocio:** Se verifica el sistema a lo largo de todo un ciclo completo de negocio.
- **Pruebas de bases de datos e integridad de datos:** Se verifica que la base de datos tenga las propiedades de atomicidad, aislamiento, durabilidad y consistencia

4.2.1.1 PRUEBAS FUNCIONALES A EJECUTAR

En la siguiente tabla se indican las pruebas funcionales a ejecutar en la solución y las pruebas funcionales que no se ejecutarán indicando el motivo por el cual no se realizarán.

Tabla 1. Pruebas Funcionales a ejecutar

TIPO DE PRUEBA	¿SE REALIZA? SI / NO	AMBIENTE EN EL QUE SE REALIZA	OBSERVACIONES
Pruebas unitarias	SI	• Ambiente local (ambiente de desarrollo)	
Pruebas de funcionalidad	SI	• Ambiente local (ambiente de desarrollo) • Ambiente pruebas interno (pruebas UT) • Ambiente de pruebas preproducción (fase 1) • Ambiente de pruebas preproducción (fase 2) • Ambiente de Producción	
Pruebas de	SI	• Ambiente local (ambiente de	

TIPO DE PRUEBA	¿SE REALIZA? SI / NO	AMBIENTE EN EL QUE SE REALIZA	OBSERVACIONES
interfaz de usuario		desarrollo) • Ambiente pruebas interno (pruebas UT) • Ambiente de pruebas preproducción (fase 1) • Ambiente de pruebas preproducción (fase 2) • Ambiente de Producción	
Pruebas de ciclo de negocio	SI	• Ambiente local (ambiente de desarrollo) • Ambiente pruebas interno (pruebas UT) • Ambiente de pruebas preproducción (fase 1) • Ambiente de pruebas preproducción (fase 2) • Ambiente de Producción	
Pruebas de bases de datos e integridad de datos	SI	• Ambiente local (ambiente de desarrollo) • Ambiente pruebas interno (pruebas UT) • Ambiente de pruebas preproducción (fase 1) • Ambiente de pruebas preproducción (fase 2) • Ambiente de Producción	

4.2.2 PRUEBAS NO FUNCIONALES

Pretenden medir el desempeño, rendimiento y disponibilidad ante fallos de los sistemas:

- **Pruebas de carga:** Se verifica que el sistema funcione adecuadamente cuando llega al límite esperado. Estas pruebas corresponden con el "Plan de pruebas de estrés y carga" de los pliegos.

- **Pruebas de estrés:** Se establecen cuáles son los umbrales extremos de funcionamiento del software una vez se supera el límite de carga esperado. Estas pruebas corresponden con el "Plan de pruebas de estrés y carga" de los pliegos.

- **Pruebas de desempeño:** Verificar los tiempos de respuesta esperados.

- **Pruebas de recuperación a fallas:** Se verifica que al forzar el fallo del software de diferentes maneras la recuperación se lleva a cabo apropiadamente.

- **Pruebas de configuración:** Se verifica que la solución se comporte de la manera esperada cuando se encuentra instalada en el ambiente de software en el que será ejecutada.

- **Pruebas de seguridad y control de acceso:** Se revisan cuáles son los riesgos de vulnerabilidad de la aplicación. Se realizan pruebas para verificar que los mecanismos de control de acceso al sistema funcionen como fueron definidos.

- **Pruebas de los servicios de interoperabilidad:** Se verifica que las diferentes aplicaciones o servicios web involucrados en la solución se relacionan adecuadamente entre sí intercambiando información y utilizando la información intercambiada.

- **Pruebas de concurrencia:** Verificar la funcionalidad del sistema al recibir una cantidad estimada de peticiones para un mismo recurso en un mismo instante de tiempo.

- **Pruebas de Ethical Hacking:** Se verifica que al realizar ataques de vulnerabilidad controlados la aplicación no es vulnerada.

4.2.2.1 PRUEBAS NO FUNCIONALES A EJECUTAR

En la siguiente tabla se indican las pruebas no funcionales a ejecutar en la solución y las pruebas no funcionales que no se ejecutarán indicando el motivo por el cual no se realizarán.

Tabla 2. Pruebas No Funcionales a ejecutar

TIPODE PRUEBA	¿SE REALIZA? SI / NO	AMBIENTE EN EL QUE SE REALIZA	OBSERVACIONES
Pruebas de carga	SI	• Ambiente de pruebas preproducción • Ambiente de Producción	
Pruebas de estrés	SI	• Ambiente de pruebas preproducción • Ambiente de Producción	
Pruebas de desempeño	SI	• Ambiente de pruebas	

TIPODE PRUEBA	¿SE REALIZA? SI / NO	AMBIENTE EN EL QUE SE REALIZA	OBSERVACIONES
		preproducción • Ambiente de Producción	
Pruebas de recuperación a fallas	SI	• Ambiente de Producción	
Pruebas de configuración	SI	• Ambiente de Producción	
Pruebas de seguridad y control de acceso	SI	• Ambiente de Producción	
Pruebas de los servicios de interoperabilidad	SI	• Ambiente de Producción	
Pruebas de concurrencia	SI	• Ambiente de pruebas preproducción • Ambiente de Producción	
Pruebas de Ethical Hacking	SI	• Ambiente de pruebas preproducción • Ambiente de Producción	

4.2.2.2 METODOLOGÍA UTILIZADA PARA LA ELABORACIÓN DE LAS PRUEBAS DE CARGA, STRESS, DESEMPEÑO Y CONCURRENCIA

Para una aplicación WEB se recomienda realizar los pasos que se indican a continuación para las pruebas de carga, stress, desempeño y concurrencia:

- PASO 1. Identificar los criterios de aceptación de desempeño.

Estos criterios están provistos por los requerimientos no funcionales que se evidencian en el plan de proyecto de la solución y se detallan en el documento de historias de usuario del proyecto. Por ejemplo:

  - El número de usuarios concurrentes que debe soportar la aplicación es cien (100)
  - El tiempo de respuesta de las páginas de consulta estadística es diez (10) segundos.

- PASO 2. Identificar los escenarios clave.

Los escenarios clave son los caminos previstos que generalmente incorporan múltiples actividades de la aplicación. Los escenarios clave son aquellos para los cuales se tienen objetivos de desempeño específicos. Son considerados de tener alto riesgo, aquellos que son más comúnmente usados, o que tienen un impacto significativo en desempeño. Los pasos básicos para identificar los escenarios clave son:

a. Identificar todos los escenarios de la aplicación WEB, un escenario normalmente puede incluir un conjunto de historias de usuario. Por ejemplo, la aplicación de Yo Cuido Lo Público puede tener los siguientes escenarios:

    - Ver detalle Elefante Blanco
    - Consultar Elefante Blanco
    - Cargar y aprobar fotos

b. Identificar las actividades involucradas en cada uno de los escenarios. Por ejemplo:

    - Un escenario de “Ver detalle Elefante Blanco” incluiría las siguientes actividades:
        - Ingresar al portal de Elefante Blanco
        - Navegar en la aplicación
        - Seleccionar un Elefante Blanco
        - Ver el detalle del Elefante Blanco
    - Un escenario de “Consultar Elefante Blanco”
        - Ingresar al portal de Elefante Blanco
        - Navegar en la aplicación
        - Seleccionar un Elefante Blanco

c. Identificar los escenarios que son los más comúnmente ejecutados o más intensivos en el uso de recursos; estos serán los escenarios claves para usar en las pruebas de carga. Por ejemplo, Consultar y aprobar Elefante Blanco, navegar por el catálogo puede ser el escenario más comúnmente ejecutado, mientras que colocar una orden puede ser el más intensivo a nivel de uso de recursos porque tiene un gran volumen de acceso a la base de datos.

- PASO 3. Crear un modelo de carga de trabajo.

El modelo de carga de trabajo o distribución es un análisis que permite identificar qué caminos de navegación van a ser realizados en la prueba y cuál va a ser la distribución de estos, es decir que carga de usuario y en qué porcentaje hará qué durante la ejecución. Cuando se está definiendo la distribución de la carga de trabajo, se deben considerar los siguientes puntos importantes para determinar las características de los escenarios de usuario.

- Un escenario de usuario es definido como un camino de navegación, incluyendo las actividades o pasos intermedios tomados por un usuario para lograr completar una tarea. También se puede pensar en ellos como una sesión.
- Un usuario típicamente hace una pausa entre las páginas durante una sesión. Esto es conocido como demora de usuario o tiempo de reflexión.
- Una sesión tendrá una duración promedio cuando es vista a través de múltiples usuarios.
- Estos datos deben ser contados cuando se están definiendo los niveles de carga que se traducirán en uso concurrente, al sobreponer usuarios o sesiones por unidad de tiempo.
- Establecer en qué proporción los caminos de navegación van a ser ejecutados unos con otros. Que caminos van a ser llevados a cabo por qué usuarios.

- PASO 4. Identificar los niveles de carga objetivo.

Identificar los niveles de carga que se aplica a la distribución carga de trabajo identificado(s) durante la etapa anterior. El propósito de la identificación de los niveles deseados de carga es asegurar que las pruebas se pueden utilizar para predecir o comparar una variedad de condiciones de carga de producción. Los siguientes son prerrequisitos comunes que se utilizan para determinar los niveles de carga de destino:

- Volumen del negocio (actual proyectado), cómo este se relaciona con los objetivos de desempeño.
- Escenarios clave
- Distribución del trabajo
- Características de sesión (camino de navegación, duración, porcentaje de nuevos usuarios)

Combinando los puntos anteriores, se pueden determinar los detalles faltantes necesarios para implementar el modelo de carga de trabajo bajo un objetivo de carga particular.

Por ejemplo, una aplicación de consulta de estadísticas puede estar proyectada para mantener un máximo de treinta (30) usuarios concurrentes. Por lo tanto, se puede modelar una prueba de carga en la que se inicia con cinco (5) usuarios y se va incrementando paulatinamente la carga hasta que se cumple con el tiempo de la prueba o el número de pruebas de rendimiento que se desean ejecutar.

- PASO 5. Identificar las métricas.

El objetivo es establecer cuáles son los indicadores clave que van a ser usados para obtener la información más efectiva que permita evaluar el rendimiento y desempeño de la solución, con el fin de determinar si existen cuellos de botella en la aplicación y si es necesario realizar ajustes sobre la aplicación. Por ejemplo, las métricas más significativas para un sistema pueden ser:

  - Tiempo promedio por página
  - Páginas por segundo
  - Errores por segundo
  - Infracciones del umbral por segundo
  - Porcentaje de tiempo de procesador
  - Uso de Memoria
  - Errores HTTP
  - Número de pruebas realizadas
  - Tiempo total de prueba
  - Tiempo duración por prueba

- PASO 6. Diseñar las pruebas específicas.

Una vez realizados los pasos anteriores es posible diseñar pruebas específicas a ser ejecutadas. Cada prueba generalmente tiene un propósito diferente, recolectar diferentes datos, incluir diferentes escenarios, y tener diferentes niveles de carga objetivo. La clave es diseñar pruebas que ayuden al equipo a recopilar la información que necesita con el fin de entender, evaluar y poner a punto la aplicación.

- PASO 7. Ejecutar las pruebas.

Ejecutar las pruebas con la herramienta definida para tal fin y recolectar los datos obtenidos para cada prueba diseñada.

- PASO 8. Analizar los resultados.

Analizar los datos recolectados y comparar los resultados contra los niveles identificados para determinar el desempeño de la aplicación.

Todos los pasos indicados en este numeral deben quedar reflejados en el informe de pruebas de carga y stress que se presente.

4.2.2.3 METODOLOGÍA UTILIZADA PARA PRUEBAS DE SEGURIDAD (VULNERABILIDAD) Y CONTROL DE ACCESO

Para una aplicación WEB se recomienda realizar los pasos que se indican a continuación para las pruebas de Seguridad (vulnerabilidad) y control de acceso:

- PASO1. Identificar perfil de la plataforma

Determinación de servicios, puertos, versiones de servicios, plataforma que soporta la aplicación, sistema operativo, versiones de bases de datos, web server, con el fin de generar un perfil de la plataforma para realizar los ataques específicos a la misma.

Las siguientes actividades se realizan dentro de este paso:

  - Escaneo de puertos
  - Identificación de sistema operativo
  - Identificación de versiones de aplicaciones y servicios
  - Identificación de servidor web
  - Identificación de tecnologías usadas en la aplicación

- PASO 2. Identificar Vulnerabilidades

Por medio de la información recolectada en el paso anterior y apoyado con herramientas específicas para la revisión de vulnerabilidades, tanto a nivel de sistema operativo, como a nivel de aplicación, se inicia una etapa donde se realiza la identificación completa de vulnerabilidades que van a ser explotadas. La identificación de las mismas se hace basada en la revisión de vulnerabilidades conocidas a nivel de parches, sistema operativo, servicios, versiones, código y

aplicación, de la misma manera que realizando peticiones no esperadas a los servicios y aplicación con el fin de determinar posibles vulnerabilidades no conocidas.

En la identificación de vulnerabilidades se realiza la búsqueda de vulnerabilidades en bases de datos, referentes a versiones y tecnologías especificas identificadas y se hace uso de herramientas automáticas de identificación de vulnerabilidades.

- PASO 3. Realizar análisis de impacto

Se realiza un análisis del posible impacto que se puede generar por la explotación de las mismas en la plataforma y se realizan las recomendaciones específicas para la mitigación o eliminación de las vulnerabilidades identificadas.

- PASO4. Realizar pruebas de revisión

Dentro de estas pruebas, se realiza una revisión de la mitigación de vulnerabilidades encontradas en la plataforma, sistema operativo y servicios ya detectadas dentro de la identificación de vulnerabilidades. Se hace una revisión completa de las vulnerabilidades y se revisan posibles vulnerabilidades asociadas a la remediación de las vulnerabilidades iniciales.

- PASO 5. Conclusiones

Se documentan las recomendaciones y conclusiones según los hallazgos y mitigaciones realizadas.

Todos los pasos indicados en este numeral deben quedar reflejados en el informe de pruebas de vulnerabilidad que se presente.

4.2.2.4 METODOLOGÍA UTILIZADA PARA PRUEBAS DE ETHICAL HACKING

Para una aplicación WEB se recomienda realizar los pasos que se indican a continuación para las pruebas de Ethical Hacking:

- PASO 1. Realizar levantamiento de Información.

Dentro de esta etapa, se busca recolectar la mayor cantidad de información respecto al sistema de información, información específica que pueda ser utilizada en la planeación de ataques, recolección de información sensible encontrada en la aplicación o los sistemas. Elementos asociados al sistema de información como equipos de comunicaciones o de seguridad perimetral que generan protección al objetivo.

Las pruebas de levantamiento de información se realizan cuando la aplicación se encuentra publicada o instalada en búsqueda de información específica de la misma.

Las pruebas incluyen las siguientes tareas

- Revisión de Registros DNS
- Revisión de Registros WHOIS
- Levantamiento de información de ruta
- Revisión de bloques de direcciones
- Búsquedas avanzadas de cabeceras y código HTML
- Análisis de metadatos
- Revisión de información pública

- PASO 2. Identificar perfil de la plataforma

Determinación de servicios, puertos, versiones de servicios, plataforma que soporta la aplicación, sistema operativo, versiones de bases de datos, web server, con el fin de generar un perfil de la plataforma para realizar los ataques específicos a la misma.

Estas pruebas incluyen la revisión a nivel interno y externo de los servidores, sistema operativo y servicios dentro habilitados en los mismos. Las pruebas se realizan a nivel de aplicación en pre producción.

Las siguientes actividades se realizan dentro de esta etapa.

- Escaneo de puertos
- Identificación de sistema operativo
- Identificación de versiones de aplicaciones y servicios
- Identificación de servidor web
- Identificación de tecnologías usadas en la aplicación

- PASO 3. Identificar Vulnerabilidades

Por medio de la información recolectada en los pasos anteriores y apoyado con herramientas específicas para la revisión de vulnerabilidades, tanto a nivel de sistema operativo, como a nivel de aplicación, se inicia una etapa donde se realiza la identificación completa de vulnerabilidades que van a ser explotadas. La identificación de las mismas se hace basada en la revisión de vulnerabilidades conocidas a nivel de parches, sistema operativo, servicios, versiones, código y aplicación, de la misma manera que realizando peticiones no esperadas a los servicios y aplicación con el fin de determinar posibles vulnerabilidades no conocidas. Las vulnerabilidades son buscadas en ambientes de pruebas y pre producción.

La identificación de vulnerabilidades se realiza ejecutando las siguientes tareas.

- Búsqueda de vulnerabilidades en bases de datos de las mismas, referentes a versiones y tecnologías especificas identificadas.

- Herramientas automáticas de identificación de vulnerabilidades

- PASO 4. Realizar explotación de Vulnerabilidades

Esta etapa se encuentra basada en la explotación de las vulnerabilidades ya identificadas dentro de la aplicación, sistema operativo o servicios, con el fin de generar condiciones que comprometan la información a nivel de confidencialidad y/o integridad y/o disponibilidad. Las pruebas buscan generar la explotación de las vulnerabilidades y la realización de ataques post explotación que permitan generar mayor compromiso a la plataforma o a la información misma. Las vulnerabilidades son explotadas buscando comprometer los tres (3) principios de seguridad de la información. Dentro de las pruebas se busca generar inicialmente compromisos a nivel de confidencialidad y/o integridad de la información, si no es posible realizar el compromiso a ese nivel, se revisan vulnerabilidades que comprometan la disponibilidad del sistema.

La explotación de las vulnerabilidades se realiza basada en un análisis de la identificación de las mismas en la etapa anterior, la explotación se realiza con scripts específicos, pruebas manuales o herramientas de generación de tráfico o peticiones maliciosas.

- PASO 5. Realizar análisis de impacto

Se realiza un análisis del posible impacto que se puede generar por la explotación de las mismas en la plataforma y se realizan las recomendaciones específicas para la mitigación o eliminación de las vulnerabilidades identificadas.

- PASO 6. Realizar pruebas de revisión

Dentro de estas pruebas, se realiza una revisión de la mitigación de vulnerabilidades encontradas en la plataforma, sistema operativo y servicios ya detectadas dentro de la realización de las pruebas iniciales. Se hace una revisión completa de las vulnerabilidades y se revisan posibles vulnerabilidades asociadas a la remediación de las vulnerabilidades iniciales.

- PASO 7. Conclusiones

Se documentan las recomendaciones y conclusiones según los hallazgos y mitigaciones realizadas, libre de vulnerabilidades asociadas a la aplicación Yo Cuido Lo Público Administrador.

Todos los pasos indicados en este numeral deben quedar reflejados en el informe de pruebas de Ethical Hacking que se presente.

4.3 ENTREGABLES DE LA EJECUCIÓN DE PRUEBAS

A continuación, se presentan los entregables de acuerdo con las pruebas que se realicen en cada ambiente:

- **Ambiente local (ambiente de desarrollo)**

Se indica en el informe de pruebas si se realizaron las pruebas unitarias según lo indicado en el ítem 4.1 de este documento.

- **Ambiente pruebas funcionales (pruebas UT)**

Los entregables de la ejecución de las pruebas son los registros de gestión de incidencias (que se indiquen en el informe de pruebas) y el informe de pruebas. Todos estos de acuerdo con las plantillas definidas para tal fin y según lo reportado en JIRA.

- **Ambiente de pruebas preproducción en versión inicial**

El resultado obtenido en la actividad de “Revisión Final de Calidad” será incluido en el informe de pruebas que se presenta como entregable en el ambiente de “Ambiente pruebas funcionales (pruebas UT)”

- **Ambiente de pruebas preproducción**

Los entregables de las pruebas realizadas en este ambiente son los registros de gestión de incidencias (que se indiquen en el informe de pruebas de la solución) y

el informe de pruebas de la Solución. Todos estos según las plantillas definidas para tal fin y lo reportado en JIRA por la Interventoría/Entidad/GEL.

- **Ambiente de Producción:**

Los entregables de las pruebas realizadas en este ambiente son:

  - Informe de pruebas de vulnerabilidad (si aplica)
  - Informe del Ethical Hacking (si aplica)
  - Informe de pruebas de estrés y carga (si aplica)

4.4 AMBIENTE DE PRUEBAS

4.4.1 HARDWARE

- El hardware que se indica a continuación corresponde con lo requerido para los ambientes

Tabla 3. Hardware requisito

RECURSO	CANTIDAD	NOMBRE / TIPO DE RECURSO
Computador analista sénior	2	• Configuración: ○ Procesador Intel Core i5 2.50 GHz. ○ 8 GB RAM. ○ 120 GB D.D. ○ Red 10/100.
Servidor Web	1	• Configuración: ○ Procesador Intel Xeon E5743 2.00 Ghz ○ 4 GB RAM. ○ 50 GB D.D. ○ Red 10/100/1000.
Servidor de base de datos	1	• Configuración: ○ Procesador Intel Xeon E5743 2.00 Ghz ○ 4 GB RAM. ○ 50 GB D.D. ○ Red 10/100/1000.

4.4.2 SOFTWARE

El software que se indica a continuación corresponde con lo requerido para los ambientes

Tabla 4. Software requisito

RECURSO	CANTIDAD	NOMBRE / TIPO DE RECURSO

Computador Analista Sénior	2	• Sistema Operativo: Windows 7.
		• Explorador(es) Web:
		  ○ Windows Internet Explorer 7.0 o superior.
		  ○ Mozilla Firefox 3.0 o superior
		  ○ Safari 5.1 o superior
		  ○ Chrome 18.0.1025.152 o superior.
		• Aplicaciones
		  ○ Visual Studio .Net 2012
		  ○ SoapUI
		  ○ MySQL Workbench

Servidor Web	1	• Servidor IIS 7.5.
Servidor de base de datos	1	• MySQL 5.6.

4.5 DESPLIEGUE DE VERSIÓN PARA DE PRUEBAS

Apoyándose en la herramienta subversión, los desarrolladores tendrán la responsabilidad del manejo de las fuentes a su cargo. La persona del equipo por demanda encargada de realizar el despliegue de las versiones se encargará de verificar que antes de pasar una versión para pruebas en cualquiera de los ambientes indicados en este documento, el código fuente, se encuentra actualizado en el respectivo repositorio y deberá crear la respectiva línea de base con la herramienta subversión teniendo en cuenta el estándar de nombramiento indicado en el plan de configuraciones que se encuentra en el repositorio 00-FABRICA DE SOFTWARE en la ruta MODULO DOCUMENTOS / Administración / 2. Planeación / Gestión de la Calidad / Plan de Configuraciones

4.6 PROCESO DE REPORTE Y MANEJO DE INCIDENCIAS

El proceso para el manejo de incidencias se realizará de acuerdo con lo indicado en el documento GLFS2-GB-OT-MetodologiaGestionDefectos, el cual se encuentra en el repositorio 00-FABRICA DE SOFTWARE en la ruta MODULO DOCUMENTOS / Administración / 2. Planeación / Proceso Gestión de Defectos

5. TERMINOLOGÍA

Apache: Es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Casos de prueba: Son un conjunto de condiciones o variables, bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

Cliente: Computador que accede a un servicio remoto en otro computador, conocido como servidor a través de una red, tiene capacidad de almacenar los datos y procesarlos, pero sigue necesitando las capacidades del servidor para una parte importante de sus funciones

Criterio de aceptación: Norma que establece la aprobación de un entregable.

Dual Core: Es un procesador que ofrece un desempeño de alto valor para ejecutar multitareas.

Equipo por demanda: Es el equipo de apoyo para los proyectos.

Fábrica de software: Línea de producción de software.

Historia de Usuario: Es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario.

MySQL: Sistema de administración relacional de bases de datos

Procesador: Encargado de ejecutar una secuencia de instrucciones almacenadas llamadas "programa". El programa es representado por una serie de números que se mantienen en una cierta clase de memoria de computador.

Servidor: Forma parte de una red, provee servicios a otras computadoras denominadas clientes, también almacena documentos HTML, imágenes, archivos de texto, escrituras, y demás material Web compuesto por datos (conocidos colectivamente como contenido), y distribuye este contenido a clientes que la piden en la red.

UTSW: Unión Temporal Software Works