

Plan de pruebas

Versión 1.1

Fecha: 25/04/2019

Autores: Steven Paniagua Aguilar, Ana Rojas Rodríguez, Silvia Calderón Navarro, José Andrés Ceciliano Granados

En el plan de pruebas se definen las estrategias de las pruebas que se le realizarán al software una vez este esté terminado.

Información específica

En esta sección se identifica el documento, a su vez describe sus orígenes e historial.

Organización emisora

En este caso cada iteración es llevada a cabo por un equipo de trabajo que consta de 4 desarrolladores, a continuación se indica el nombre completo de cada desarrollador:

- Steven Paniagua Aguilar
- Ana Rojas Rodríguez
- Silvia Calderón Navarro
- José Andrés Ceciliano

Autoridad de aprobación

Las personas encargadas de revisar y firmar este documento son los mismos 4 desarrolladores que se mencionan en el punto anterior:

- Steven Paniagua Aguilar
- Ana Rojas Rodríguez
- Silvia Calderón Navarro
- José Andrés Ceciliano

Historial de cambios

Fecha	Versión	Cambio	Persona
30/03/2019	1.0	Esta es la primer versión del documento, por lo que no se poseen documentos predecesores.	<ul style="list-style-type: none">• José André Ceciliano• Ana Rojas• Silvia Calderón• Steven Paniagua.
24/04/2019	1.1	Esta es la segunda versión del documento. Se agregan nuevos requerimientos.	<ul style="list-style-type: none">• José André Ceciliano• Ana Rojas• Silvia Calderón• Steven Paniagua.

Introducción

En esta sección se proporciona información explicativa sobre el contexto y la estructura del documento.

Alcance

El propósito de este documento es proporcionarle la información y el marco para darnos paso a realizar todos los procesos de prueba en el producto de software que consiste en una red social llamada Face-TEC

Referencias

- Plan de Proyecto
- Especificación de requisitos del proyecto
- Especificación de hardware y software
- Diseño del producto

Contexto de las pruebas.

Proyecto

Este proyecto fue desarrollado implementando el siguiente lenguaje y herramientas:

- PHP
- Apache 2
- MySQL

- Vesta CP

La arquitectura en la que se desarrolló es la siguiente:

- Server 1GB RAM
- 50 SSD
- Ubuntu 16.04

Prueba de artículo (s)

Para poder verificar la perfecta funcionalidad de este proyecto, se realizarán pruebas dinámicas a:

- El registro de usuarios en la red social.
- El logeo de usuarios en el sistema.
- La creación y publicación de post, además de su posible visualización.
- La realización de comentarios.
- Darle like al post de un amigo o públicos y visualizar la lista de usuarios que han dado like a un post.
- La búsqueda de post por medio de los tags.
- Agregar imágenes al perfil.
- Agregar imágenes a los post.
- Fotos en los comentarios.
- Comentar un comentario.
- Darle like a una respuesta de un comentario en específico.
- La robustez del sistema.

En el caso de las pruebas estáticas se harán pruebas al:

- Código del producto de software desarrollado, verificando que cumpla con el principio de clean code.
- Estándar PHP establecido por el equipo , determinando que se esté aplicando de forma correcta en el proyecto.

Además cada versión exacta con respecto a los diversos elementos de prueba deben ser obtenidos por el sistema de administracion de configuracion, esto al momento de especificar la prueba, teniendo en cuenta que deben ser controlados antes de ejecutar alguna de las pruebas establecidas.

Alcance de la prueba

En este caso se realizarán pruebas de cada uno de los requerimientos establecido, esto basándose en que el sistema operativo y la red donde se van a llevar a cabo las pruebas funcionan de forma correcta, esto con el fin de no tener inconvenientes al momento de ejecutar una prueba. También se probarán factores como lo son la robustez y su facilidad de uso. Por último se planea realizar pruebas programadas que se encarguen de verificar en buen funcionamiento del sistema web en su totalidad.

Suposiciones y Restricciones

Es importante mencionar que cada miembro del equipo tendrá libre acceso al código fuente del producto, así como a la base de datos empleada por este mismo. Esto con el fin de que todos puedan llevar a cabo las pruebas establecidas sin ningún inconveniente.

Cabe destacar que con las pruebas estáticas el encargado de llevarlas a cabo se basa en los estándares previamente establecidos por el equipo, lo cuales son:

- Clean Code
- PSR-4 COPEL

En el caso de las pruebas dinámicas cada miembro del equipo debe tener en cuenta los requerimientos establecidos para el proyecto:

- El registro de usuarios en la red social.
- El logeo de usuarios en el sistema.
- La creación y publicación de post público y privados, además de su posible visualización, los privados sólo pueden ser visualizados por amigos.
- La realización de comentarios.
- Darle like al post de un amigo o públicos y visualizar la lista de usuarios que han dado like a un post.
- La búsqueda de post por medio de los tags.
- Registro de imágenes en el perfil.
- Imágenes en los post.
- Comentarios con fotos.
- Comentar un comentario.
- Darle like a una respuesta de un comentario.
- La robustez del sistema y la facilidad de uso.

Partes interesadas

Nombre	Cargo
Ana Rojas Rodriguez	Desarrolladora
Steven Paniagua Aguilar	Desarrollador
Silvia Calderón Navarro	Desarrolladora
José Ceciliano Granados	Desarrollador

Comunicación de prueba

En este caso la comunicación del equipo de desarrollo del proyecto se llevará a cabo por

medio de reuniones semanales ya sea en forma presencial o videollamada, esto con el fin de determinar el estado y avance del producto de software que se encuentran desarrollando, por otro lado, cada miembro del equipo tendrá una tarea referente a las pruebas, ya sea resolver el problema, llevar a cabo la prueba o bien revisar el resultado obtenido de esta. A continuación se define a cada encargado de llevar a cabo las partes que conforman la prueba.

Miembro del equipo	Encargado de programar pruebas	Encargado de resolver problemas	Encargado de realizar la prueba	Encargado de revisar la prueba
Ana Rojas Rodriguez	X		X	
Steven Paniagua Aguilar	X		X	
Silvia Calderón Navarro	X			X
José Ceciliano Granado	X	X		

Registro de riesgo

A continuación se muestran las abreviaturas que serán empleadas en la tabla de riesgos, esto con el fin de que se tenga una idea más clara de qué significa cada una de ellas:

- P = Probabilidad del riesgo
- I = Impacto que genera el riesgo si este sucede
- E = Exposición = Probabilidad x Impacto

En este caso la escala de probabilidad e impacto será de 1-6, siendo 1 la más baja y 6 la más alta.

Riesgos del producto

ID	Riesgo	P	I	E	Actividades de mitigación
1	Es imposible registrar usuarios	2	6	12	<ul style="list-style-type: none"> • Analizar el código que se encarga de tomar los datos y registrarlos. • Verificar que la base de datos funciona de forma correcta y no esté impidiendo el ingreso de esos datos en las tablas.

2	No es posible loguearse en el sistema.	2	5	10	<ul style="list-style-type: none"> • Analizar la parte de código que valida los datos de entrada. • Verificar que la base de datos esté devolviendo los datos para poder validar.
3	No es posible registrar un post.	4	6	24	<ul style="list-style-type: none"> • Verificar que el código encargado no tenga algún tipo de anomalía. • Determinar si el problema radica en la base de datos y su posible solución.
4	No es posible visualizar un post publicado.	3	6	18	<ul style="list-style-type: none"> • Verificar el código fuente en búsqueda de algun problema de programacion. • Analizar si el problema es en la interfaz y si existe una solución. • También el problema podría ser en la base de datos por lo que es recomendable revisar detalladamente.
5	No se pueden realizar comentarios en los post de amigos o los que son públicos.	3	6	18	<ul style="list-style-type: none"> • Verificar que el código fuente no presente algún error. • Verificar la existencia de errores en la interfaz. • Verificar el buen funcionamiento de la base de datos.
6	No se pueden visualizar los comentarios hechos a un post.	2	5	10	<ul style="list-style-type: none"> • Analizar la parte de código que muestra esos datos. • Verificar que la base de datos esté devolviendo los datos para poder mostrarlos. • verificar que no hay un inconveniente en la interfaz que evita desplegar los datos.
7	No es posible darle like a un	3	6	18	<ul style="list-style-type: none"> • Verificar anomalías en

	post publicado.				<p>la interfaz y código fuente de esta.</p> <ul style="list-style-type: none"> • Determinar que no hayan errores en el algoritmo que hace el like.
8	No se pueden visualizar los likes hechos a un post.	2	5	10	<ul style="list-style-type: none"> • Analizar la parte de código que muestra esos datos. • Verificar que la base de datos esté devolviendo los datos de forma correcta. • verificar que no hay un inconveniente en la interfaz que evita visualizarlo.
9	No se pueden buscar los post por medio de sus tags.	3	5	15	<ul style="list-style-type: none"> • Determinar si el algoritmo de búsqueda funciona de forma correcta y es eficiente. • Verificar si el problema es a raíz de la interfaz o la base de datos.
10	Los post públicos se le muestran a todos los usuarios del sistema.	2	6	12	<ul style="list-style-type: none"> • Verificar que el algoritmo que determina si un post es público o privado esté funcionando de forma correcta.

Riesgos del proyecto

ID	Riesgo	P	I	E	Actividades de mitigación
1	Mala comunicación en el equipo de desarrollo.	3	6	18	<ul style="list-style-type: none"> • Realizar reuniones semanales con el equipo. • Mantener un clima de confianza entre los integrantes.
2	Mala planificación del tiempo	2	4	8	<ul style="list-style-type: none"> • Durante la planificación se debe tomar en cuenta posibles retrasos en el desarrollo y sus posibles consecuencias.

					<ul style="list-style-type: none"> • Establecer más horas de las que se necesitan para terminar el proyecto, esto en caso de un atraso.
3	Personal insuficiente	2	3	6	<ul style="list-style-type: none"> • Estimar correctamente el esfuerzo del proyecto. • Seguir el progreso de la prueba detenidamente. • Informar claramente el progreso y los problemas de recursos en cada oportunidad que se presente.
4	Poca experiencia en algunos miembros del equipo	2	4	8	<ul style="list-style-type: none"> • Capacitar a los miembros que requieren más conocimiento en el proyecto. • Incluir un entrenamiento en el horario. • Revisar cuidadosamente el trabajo hecho por los individuos sin experiencia.
5	Algunos de los integrantes no pueden trabajar juntos	5	2	10	<ul style="list-style-type: none"> • Identificar el problema entre los individuos. • Tratar de solucionar la situación entre los individuos. • Si no es posible llegar a una solución, asignar actividades en las cuales no haya contacto entre ellos.
6	Uno de los miembros abandona el equipo.	2	4	8	<ul style="list-style-type: none"> • Determinar la razón por la que abandona el equipo y ver si es posible resolver la situación. • De no ser así, se distribuye el trabajo que dejó entre los demás miembros en partes iguales.

7	Mala comunicación con el cliente	3	5	15	<ul style="list-style-type: none"> • Llevar a cabo reuniones semanales con el cliente. • Indicarle el avance del proyecto en cada reunión.
8	Cambios inesperados en el producto por parte del cliente.	5	3	15	<ul style="list-style-type: none"> • Llevar un buen control del avance para determinar con exactitud dónde aplicar los cambios sin afectar ese avance. • Verificar que los cambios están dentro del alcance.
9	Tiempo insuficiente para realizar las pruebas.	3	6	18	<ul style="list-style-type: none"> • Reservar en el cronograma un tiempo prudente a las pruebas. • Asignar el personal suficiente para que lleven a cabo las pruebas.

Estrategia de prueba

En esta sección se describe el enfoque de las pruebas para el proyecto de prueba especificado o el subproceso de prueba.

Probar los subprocesos

Las pruebas establecidas para este producto de software incluyen los siguientes subprocesos de prueba:

- **Revisiones Técnicas:** Este es un tipo de revisión del estándar cuyo objetivo, es reconocer tempranamente defectos para evitar que se extiendan a otros espacios del código. Esta es la forma en la que se realizarán las pruebas estáticas documentando cualquier problema que exista con el código o el estándar php.
- **Pruebas de Componentes:** Consiste en probar los elementos independientes del sistema y entre las características que se pueden medir se encuentra la robustez, especificada anteriormente como requerimiento.
- **Pruebas de integración:** En este caso se llevarán a cabo pruebas en la interfaz del producto, determinando el correcto funcionamiento de cada componente de esta, ya sea que muestre los datos de forma correcta y funcione de acuerdo a los requerimientos, así como que cumpla con los

estándares establecidos.

- **Pruebas de sistema:** Con este tipo de revisión se espera determinar el buen funcionamiento del modelo y controlador, verificando el registro correcto de datos y su posible acceso, todo por medio de la interfaz.
- **Pruebas de aceptación:** Consisten en todas las normativas que deben cumplirse, en este caso se pueden considerar como normativas el estándar PHP seleccionado por el equipo de trabajo, además de la especificación del producto a desarrollar
- **Pruebas programadas:** Son pruebas hechas en el entorno PHPunit con el fin de verificar el buen funcionamiento del sistema en base a lo que se encuentra programado en el lenguaje PHP respectivamente.
- **Arreglo ortogonal:** Es una técnica que permite encontrar los efectos de ciertas variables presentes en el proyecto que el equipo se encuentra desarrollando, de este modo predecir posibles condiciones futuras de operación. Las variables elegidas son el Sistema Operativo, el navegador, la búsqueda de post y la cantidad de amigos.

Resultados de prueba

Para cada subprocesos de prueba se debe producir la siguiente documentación:

- **Minuta:** constituye un pequeño informe en el caso de las pruebas estáticas, que debe contener la siguiente información:
 - Responsable de llevar a cabo la prueba.
 - La descripción de la prueba a realizar.
 - El objetivo de la prueba.
 - La fecha.
 - El resultado obtenido.
 - La reevaluación de riesgos.
 - Tipo de prueba.
- **Registro de ejecución de pruebas:** es una hoja de cálculo con la información de las pruebas dinámicas y programadas, cada prueba debe tener los siguientes puntos:
 - Eventos
 - Identificador único
 - Tiempo
 - Descripción
 - Impacto

Técnicas de diseño de prueba

En este caso las técnicas de calidad de software que será empleadas por el equipo

de trabajo son:

- **Tormenta de ideas:** Consiste en una técnica que se realiza en grupos, intentando estimular a cada miembro sin complejos en su aportación de ideas que pueden resolver una determinada situación.
- **Diagramas de sectores:** Esta técnica emplea un círculo dividido en varios sectores usado en todo tipo de variables, generalmente cualitativas.

Criterios de finalización de la prueba

Las siguientes condiciones muestran el momento en el que la etapa de prueba de un requerimiento, está completa y por ende la aplicación de la prueba ya no será necesaria:

- Para las pruebas de componentes las condiciones a cumplir serán:
 - Registro de 10 usuarios y que no permita que ningún nombre de usuario sea igual al de otro usuario ya registrado.
 - El logueo efectivo de unos 5 usuarios registrados y el no posible logueo de 5 usuarios sin registrar en el sistema.
 - Creación de 10 post y su debida publicación.
 - Realización de 5 comentarios en un post y su posible visualización.
 - Al darle like a 10 post, 5 privados y 5 públicos, esto se puede apreciar en la interfaz.
 - Registrar 10 tags a un comentario y se pueden apreciar en la interfaz.
 - Buscar post's por medio del tag registrado en el comentario, y los resultados se muestran en pantalla.
- Las pruebas de integración se cumplen de la siguiente forma :
 - Realización de 8 comentarios con diferentes usuarios, 5 de esos comentarios son respuesta a un post con diferentes usuarios, los otros 3 a uno privado, eliminar amigo con post privado. Todo comentario público debe mostrarse y ningún comentario de un post privado debe enseñarse a usuarios que no son amigos del usuario que posee sus post privados.
 - Dar like a todos los post de un amigo, visualizar la lista de usuarios que le dieron like, eliminar a uno de los usuarios que dio like como amigo, volver a buscar el post y visualizar los like's. En este caso debe aparecer el like del usuario que eliminó al amigo.
 - Buscar post y comentarios por medio de tag, visualizar sus like's y darle like.
 - Probar la robustez de la aplicación registrando 10 usuarios con caracteres especiales y que todos estos se puedan loguear.
- Las pruebas estáticas serán completas cuando cumpla con lo siguiente:
 - Luego de un análisis detallado del código presente en la secciones de Modelo, Vista, Controlador y todo este cumpla el principio Clean Code y el estándar PHP establecido.
- El arreglo ortogonal está completo cuando cumpla con 25 búsquedas de post, combinando cada variable determinada.

- Las condiciones que deben cumplir las pruebas de sistema son:
 - Análisis completo de la interfaz, determinando que cada requerimiento establecido es aplicado en esta y funciona de forma correcta. Una sola prueba en la que se registra un usuario, se loguea, crea un post público y otro privado, hacer un amigo, le da like a sus post y busca un post por medio de los tags. Se trata de medir el flujo normal de la aplicación.
- Las pruebas de aceptación estarán completas cuando se consuma el siguiente requisito:
 - Verificar las pruebas encargadas de revisar el Estándar PHP y que todas estén completas. El objetivo de esta prueba es la aceptación de regulación, debido a que el Estándar PHP, los principios Clean Code y la especificación son los documentos más cercanos a normas del proyecto.
- Las pruebas programada deben cumplir:
 - Loguearse con parámetros y sin parámetros.
 - Loguearse con un pass correcto e incorrecto.
 - Realizar publicaciones de post sin parámetros y con parámetros.
 - Realizaron comentarios con y sin archivo.
 - Agregar un amigo.
 - Eliminar un amigo.
 - Realizar un like.
 - Realizar un dislike.
 - Like a un comentario.
 - Dislike a un comentario
 - Tags con datos y sin datos.
 - Obtener comentarios.
 - Obtener la lista de amigos.
 - Obtener los likes de post y comentarios.
 - Funcionamiento del botón de like para post y comentarios.
 - Obtener información del perfil(nombre, imagen, usuarios).
 - Eliminar datos.

Métricas a recoger

- Verificar el cumplimiento del código limpio.
- Determinar la correcta aplicación del estándar PHP establecido.
- Verificar los efectos de las siguientes variables en el sistema:
 - Sistema Operativo
 - Navegador
 - Post
 - Amigos
- Comprobar los requerimientos establecidos:
 - El registro de usuarios en la red social.
 - El logeo de usuarios en el sistema.
 - La creación y publicación de post público y privados, además de su

posible visualización, los privados sólo pueden ser visualizados por amigos.

- La realización de comentarios.
- Darle like al post de un amigo o públicos y visualizar la lista de usuarios que han dado like a un post.
- La búsqueda de post por medio de los tags.
- Agregar imágenes al perfil.
- Agregar imágenes a los post.
- Fotos en los comentarios.
- Comentar un comentario.
- Darle like a una respuesta de un comentario en específico.
- La robustez del sistema y la facilidad de uso.

Requisitos de datos de prueba

No existe ningún requisito en los datos de prueba, es decir, las pruebas no están sometidas a ninguna restricción.

Requisitos del entorno de prueba

Para poder llevar a cabo cada prueba en el producto de software, el encargado de realizarla debe tomar en cuenta que requiere de las siguientes herramientas para poder ejecutar el proyecto:

- Apache 2
- MySQL
- Vesta CP
- PHPunit

También las pruebas se llevan a cabo bajo la siguiente arquitectura:

- Server 1GB RAM
- 50 SSD
- Ubuntu 16.04 ó Windows 10

Criterios de suspensión y reanudación.

Los criterios de suspensión se pueden apreciar en la tabla de riesgos del proyecto, cabe destacar que si algunos de estos riesgos se materializan, se suspenderán las pruebas, y solo hasta que se resuelva cada inconveniente presente en el proyecto se podrán reanudar dichas pruebas en su debido tiempo.

Desviaciones de la estrategia de prueba organizacional

No existe ninguna plan de desviación de la estrategia de prueba organizacional, sin embargo, todo el equipo de desarrollo es el encargado de planificar y aprobar un plan de desviación de ser necesario en un futuro.

Actividades y estimaciones de pruebas.

Las actividades de prueba es aplicar todas las pruebas establecidas en este plan de pruebas. Es sección permite informar sobre las pruebas, el resultado esperado y determinar el responsable de llevar a cabo cada una de las pruebas, así como de revisarlas y aprobarlas. Esto con el fin de mantener un orden a la hora de llevar a cabo cada actividad de prueba. Las siguientes son las actividades de prueba:

- Pruebas Estáticas
- Pruebas Dinámicas:
 - Pruebas de componentes.
 - Pruebas de integración.
 - Pruebas de sistema.
 - Pruebas de aceptación.
- Pruebas Programadas

Dotación de personal

Para poder llevar a cabo cada una de las pruebas, los miembros del equipo deben tener el mayor conocimiento posible del estándar establecido en un inicio, así como de los requerimientos del producto en desarrollo

Roles, actividades y responsabilidades

Miembro del equipo	Pruebas de integración	Pruebas estáticas	Pruebas de Componentes	Pruebas de sistema	Pruebas de aceptación	Pruebas programadas
Ana Rojas Rodriguez	Aprobación			Realiza	Revisión Aprobación	Realiza Revisión
Steven Paniagua Aguilar	Revisión	Realiza	Aprobación		Realiza	Realiza Aprobación
Silvia Calderón Navarro	Realiza	Aprobación	Revisión			Realiza Aprobación
José Ceciliano Granado		Revisión	Realiza	Aprobación Revisión		Realiza Revisión