

UF2. [PAC01] Solución

Para la realización de las siguientes cuestiones deberéis ayudaros del libro de texto, capítulos 8, 9 y 10, y consultar, si lo creéis necesario, internet.

Debéis subir un único archivo comprimido que contenga UN ÚNICO DOCUMENTO PDF con las respuestas del test y de las preguntas cortas, y además se adjuntarán las soluciones de la parte práctica, con nombres identificativos.

La pregunta con el tipo test tiene un valor de 2 puntos, por lo que las preguntas tipo test tienen un valor de 0,2 cada una. El resto de preguntas cortas de teoría vale 3 puntos cada una. Por último, el ejercicio práctico vale 2 puntos.

Las preguntas tipo test se corregirán sólo en la caja que veréis después de las preguntas.

Recordad que la fecha límite para la entrega de la PAC es el día 11 de ENERO.

Sugerencia: Para la realización del ejercicio práctico es necesario tener instalado Visual Studio.

Información: Cualquier PAC copiada y/o en la que se haya utilizado “copy-paste” de código ya escrito será puntuada con un 0.

Actividades

Parte teórica

1. Responde a este test en la matriz de respuestas que se encuentra al final de las preguntas:
 - 1.1. Para poder ejecutar los comandos javac debemos modificar una variable de sistema ¿Cuál?
 - a. PATH
 - b. CATALINA_HOME
 - c. JAVA_HOME
 - d. Ninguna de las anteriores

- 1.2. Para compilar un fichero .java debemos usar el comando:
- java
 - javac
 - jar
 - jre
- 1.3. Las pruebas de caja blanca cubren campos tales como:
- Cobertura de sentencias, cobertura de decisiones
 - Cobertura de condiciones
 - Cobertura de caminos.
 - Todas las anteriores.
- 1.4. Las pruebas de caja negra cubren campos tales como:
- Cobertura de sentencias, cobertura de decisiones
 - Cobertura de condiciones
 - Cobertura de caminos.
 - Ninguna de las anteriores.
- 1.5. Podemos definir prueba de sistema como:
- Aquellas que buscan asegurar que el código de acuerdo con las especificaciones definidas y que el módulo lógico es válido. La operativa de este tipo de pruebas se centra en ejecutar cada módulo (o unidad mínima a ser probada, como por ejemplo una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores.
 - Aquellas en las que se prueban los diferentes módulos o componentes de forma conjunta a fin de encontrar errores que pudieran ser producidos por los efectos de uno sobre otro.
 - Aquellas que tienen como objetivo testear de forma completa todo el sistema, de forma que compruebe la correcta integración de todos los elementos y que opera de forma correcta de acuerdo con las especificaciones definidas.
 - Aquellas que consisten en la repetición selectiva de pruebas para detectar fallos introducidos durante la modificación de un sistema o componente del sistema.

1.6. Podemos definir prueba de regresión como:

- a. Aquellas que buscan asegurar que el código de acuerdo con las especificaciones definidas y que el módulo lógico es válido. La operativa de este tipo de pruebas se centra en ejecutar cada módulo (o unidad mínima a ser probada, como por ejemplo una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores.
- b. Aquellas en las que se prueban los diferentes módulos o componentes de forma conjunta a fin de encontrar errores que pudieran ser producidos por los efectos de uno sobre otro.
- c. Aquellas que tienen como objetivo testear de forma completa todo el sistema, de forma que compruebe la correcta integración de todos los elementos y que opera de forma correcta de acuerdo con las especificaciones definidas.
- d. Aquellas que consisten en la repetición selectiva de pruebas para detectar fallos introducidos durante la modificación de un sistema o componente del sistema.

1.7. Podemos definir prueba alfa como:

- a. Aquellas que tienen por objetivo probar que el sistema desarrollado cumple con las funcionalidades específicas para las que ha sido creado. En este tipo de pruebas participan analistas y usuarios finales.
- b. Aquellas que están encaminadas a ver cómo responde el sistema en ciertas condiciones de trabajo, elevado número de usuarios, conexiones simultáneas a la base de datos, etc. Es decir, simula el comportamiento de la aplicación.
- c. Aquellas que se llevan a cabo en el entorno del desarrollador, mediante un grupo representativo de usuarios finales, esto es, en un entorno controlado bajo la supervisión del desarrollador.
- d. Aquellas que se realizan cuando el producto está listo para implantarse en el entorno cliente. Estas pruebas deben ser realizadas por usuarios finales.

- 1.8. Podemos definir prueba beta como:
- Aquellas que tienen por objetivo probar que el sistema desarrollado cumple con las funcionalidades específicas para las que ha sido creado. En este tipo de pruebas participan analistas y usuarios finales.
 - Aquellas que realiza el usuario final en un entorno no controlado y sin la supervisión del desarrollador.
 - Aquellas que se llevan a cabo en el entorno del desarrollador, mediante un grupo representativo de usuarios finales, esto es, en un entorno controlado bajo la supervisión del desarrollador.
 - Aquellas que se realizan cuando el producto está listo para implantarse en el entorno cliente. Estas pruebas deben ser realizadas por usuarios finales.
- 1.9. Los elementos textuales que aparecen en la interfaz al posicionarse el usuario con el puntero del ratón durante un par de segundos sobre el elemento se conoce como:
- HelpTip
 - ToolTip
 - ElemTip
 - Ninguna de las anteriores.
- 1.10. Durante el desarrollo de un proyecto de software se generan los siguientes tipos de documentación:
- Manual de instalación, Manual de usuario, Manual de configuración
 - Guía de usuario, guía rápida
 - Manual de administración
 - Todas las anteriores.

MATRIZ DE RESPUESTAS

1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10
A	B	D	D	C	D	C	B	B	D

2. Realiza una descripción detallada de los tipos de pruebas existentes. Puedes ayudarte de tablas, esquemas, dibujos, diagramas...

PRUEBA UNITARIA

Buscan asegurar que el código funciona de acuerdo con las especificaciones definidas y que el módulo lógico es válido. La operativa de este tipo de pruebas se centra en ejecutar cada módulo por separado lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores.

Descripción de la prueba:

- Particionar los módulos en pruebas en unidades lógicas fáciles de probar.
- Por cada unidad hay que definir los casos de prueba, que serán de caja blanca.
- Para esto, los casos de prueba deben diseñarse de tal forma que se recorran todos los caminos de ejecución posibles dentro del código bajo prueba.
- Los aspectos a considerar son los siguientes:
 - Rutinas de excepción
 - Rutinas de error
 - Manejo de parámetros
 - Validaciones
 - Valores válidos
 - Valores límites
 - Rangos
 - Mensajes posibles
- Técnicas
 - Comparar el resultado esperado con el resultado obtenido.

PRUEBA DE INTEGRACIÓN

Consisten en probar todos los componentes de una aplicación de forma conjunta. Se suele realizar esta combinación de forma incremental, pueden ser ascendente o descendente:

- Ascendente:
 - Se combinan los módulos de bajo nivel en grupos que realicen una subfunción o tarea específica.
 - Se diseña un gestor para coordinar la entrada y salida de los casos de prueba.

- Se comprueba el funcionamiento del grupo.
- Se eliminan los gestores y se combinan los grupos moviéndose hacia arriba por la estructura del programa.
- Descendente:
 - Se usa el módulo de control principal como controlador de la prueba, creando resguardos para todos los módulos directamente subordinados al módulo de control.
 - Dependiendo del enfoque e integración elegido se van sustituyendo uno a uno los resguardos subordinados por los módulos reales.
 - Se llevan a cabo pruebas cada vez que se integra un nuevo módulo.

PRUEBA DE SISTEMA

Este tipo de pruebas tiene como objetivo testear de forma completa el sistema, de forma que se compruebe la correcta integración de todos los elementos (hardware y software) y que opera de forma correcta de acuerdo a las especificaciones definidas. El objetivo de la prueba es realizar los siguientes puntos, en donde, a diferencia de otras pruebas que se centran en un aspecto concreto, se evalúan diferentes aspectos:

- Se cumplen los requisitos funcionales definidos.
- El funcionamiento y el rendimiento de las interfaces hardware, software y de usuario.
- La adecuación de la documentación de usuario.
- Rendimiento y respuestas en condiciones límite y de sobrecarga.
- Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación.

Para la generación de pruebas, al igual que en el caso anterior, se utilizan pruebas de caja negra. Este tipo de pruebas se realizan inicialmente en el entorno de desarrollo, denominadas pruebas alfa, posteriormente en el entorno cliente, denominadas pruebas beta.

Las pruebas de sistema en las que se busca testear el sistema en diferentes entornos de configuración de hardware y software se denominan pruebas de configuración. En esta prueba se determina que pruebas de sistema aseguran que la aplicación alcanzará sus objetivos de negocio.

Estas pruebas incluyen:

- De funcionalidad
- De usabilidad

- De performance
- De múltiples sitios
- De documentación y procedimientos
- De seguridad y controles
- De volumen
- De esfuerzo
- De recuperación

PRUEBAS DE REGRESIÓN

Cada vez que se agrega un nuevo módulo se introducen nuevas rutas de flujo de datos y una nueva ruta lógica de control de software. Estos nuevos cambios pueden producir errores en módulos que anteriormente funcionaban de forma correcta. Por lo tanto es necesario comprobar que los nuevos cambios no han producido efectos colaterales negativos.

La regresión consiste en la repetición selectiva de las pruebas para descartar fallos introducidos durante la modificación.

En las pruebas de regresión hay que:

- Probar íntegramente los módulos que han cambiado.
- Decidir las pruebas a efectuar para los módulos que no han cambiado y que han sido afectados por los cambios efectuados.

Este tipo de pruebas ha de realizarse tanto durante la fase de desarrollo como cuando se produzcan cambios en el software, incluyendo en este apartado el mantenimiento.

PRUEBAS FUNCIONALES

Tiene por objetivo, probar el sistema desarrollo, cumpla con la funcionalidad específica para la que ha sido creado. En este tipo de pruebas se emplean equipos formados por analistas de pruebas y usuarios finales. Se realiza en la fase final de proyecto y permite decidir si pasar al entorno de producción en caso de ser satisfactoria.

PRUEBAS DE CAPACIDAD Y RENDIMIENTO

Destinadas a ver cómo responde el sistema en condiciones de trabajo. Las pruebas pueden ser:

- Humo, para cargas ligeras de trabajo.
- Tensión, para cargas pesadas durante tiempo continuado.

- Rendimiento, determina como corresponde la aplicación.
- Diseño de capacidad, determina cómo se comporta la aplicación bajo distintas capacidades.

En cuanto al proceso de carga de datos podemos ver diferentes modelos de cómo se facilitan estos datos como se realiza la simulación. Existen diferentes modelos de carga:

- Constante: se establece la misma carga de un usuario durante un tiempo constante.
- Paso: permite supervisar la aplicación a medida que se aumentan los valores de carga. Permite conocer los umbrales en los que los tiempos de respuesta de la aplicación se encuentran dentro de los parámetros aceptables.
- Basado en objetivos: similar a la anterior, pero además permite determinar cuándo detener el aumento de carga.

PRUEBA DE USO DE RECURSOS

Esta prueba determina que elementos son críticos para el sistema, siendo habitualmente la memoria RAM, espacio en disco y CPU.

PRUEBAS DE SEGURIDAD

Estas pruebas tratan de asegurar que los mecanismos definidos para evitar accesos no autorizados funcionan correctamente.

PRUEBAS DE USUARIO

Permiten comprobar la usabilidad de la aplicación. Se basan en la observación de las técnicas de navegación, acceso e interacción que emplean diferentes usuarios al acceder a la aplicación.

Se escogen grupos de usuarios con diferentes perfiles y habilidades en el uso de aplicaciones informáticas. Se mide la manera en la que el usuario hace una tarea determinada, cuanto tiempo emplea, número de clics y errores que comete durante el proceso. Existen muchas formas de verificar la facilidad de uso. Los ensayos incluyen:

- Encuestas y entrevistas.
- Grupos de enfoque.
- Pruebas de laboratorio con determinadas herramientas y software avanzado.
- Evaluaciones heurísticas realizadas por expertos en usabilidad.

En líneas generales se comienza estableciendo un propósito general para la investigación, manifestándolo en unos pocos objetivos para diseñar cada test.

Durante el transcurso de las pruebas uno o varios moderadores observan en silencio y toman notas acerca de la navegación del usuario.

PRUEBA DE ACEPTACIÓN

A la hora de realizar estas pruebas, el producto está listo para implantarse en el entorno del cliente. El usuario debe ser el que realice las pruebas, ayudado por personas del equipo de pruebas, siendo el propio usuario el que aporte casos de prueba.

Estas pruebas se caracterizan por:

- Participación activa del usuario, que debe ejecutar los casos de prueba ayudado por el equipo.
- Están enfocadas a probar los requisitos de usuario.
- Corresponden a la fase final del proceso de desarrollo.
- Es muy recomendable que las pruebas de aceptación se realicen en el entorno en que se va a explotar el sistema.

Pruebas alfa y beta

- Prueba alfa: Se lleva a cabo en el entorno del desarrollador, mediante un grupo representativo de usuarios finales, esto es, en un entorno controlado bajo la supervisión del desarrollador.
- Prueba beta: El usuario carece de la supervisión mencionada y se encuentra en un entorno no controlado, similar al que tenía en producción.

3. Realiza una descripción detallada de los tipos de documentos que suelen acompañar a la creación de un proyecto. Puedes ayudarte de tablas, esquemas, dibujos, diagramas...

MANUAL DE INSTALACIÓN:

Contiene todos los detalles necesarios para llevar a cabo la instalación de la aplicación en diferentes entornos. Se contemplan en esta documentación aspectos referidos al software como pueden ser diferentes sistemas operativos, otras aplicaciones instaladas dependencias de otros programas y aspectos referidos a características mínimas de hardware.

Se explican también diferentes posibilidades de instalación:

- Cambio de directorios por defecto.
- Generación o no de archivos de log.

- Ubicación archivos de configuración.
- Tipo de instalación.
- Apariencia.

De esta documentación suele haber dos versiones:

- Reducida: para la instalación rápida, sin profundizar en demasiados aspectos.
- Detallada: se explican ampliamente cada uno de los aspectos anteriormente nombrados.

DOCUMENTACIÓN DE CONFIGURACIÓN

Dedicada a explicar todos los parámetros de configuración de la aplicación. En ella se mezclan aspectos de software, hardware, visuales, etc.

Debe especificar como realizar cambios en la configuración así como las consecuencias que cada una de estas modificaciones tiene sobre el programa.

Habitualmente se pueden encontrar varios niveles de configuración:

- Normal
- Avanzada

Estos dos niveles de configuración, en ocasiones, se traducen en diferentes versiones de guías de configuración, una para cada nivel.

MANUAL DE USUARIO

Existen dos versiones, la rápida y la detallada. La diferencia radica en el nivel de profundidad con el que se tratan las diferentes posibilidades de la aplicación.

En este tipo de documentación, hay que tener en cuenta que el nivel de los usuarios que las usan no es homogéneo y en algunos casos será necesario realizar varias versiones:

- Principiante.
- Medio.
- Avanzado.
- Experto.

Dependiendo del nivel se profundizará más o menos en las explicaciones, centrándose los niveles superiores en explicar la funcionalidad del programa, desde un punto de vista práctica, sin redundar en detalles técnicos. Estos detalles aumentan a medida que el nivel del manual aumenta.

GUÍA DE USUARIO

Tiene como objetivo indicr de froma rápida y sencilla ciertas funcionalidades del producto de forma que el cliente pueda operar de forma inmediata.

GUÍA RÁPIDA

Detalla de forma rápida y muy sintetizada la forma de realizar una tarea.

MANUAL DE ADMINISTRACIÓN

Permite conocer los detalles de la instalación, la administración del sistema y la configuración de los diferentes componentes de la aplicación o sistema. Este tipo de manuales son utilizados por los administradores.

Parte práctica

4. Crea una prueba unitaria que evalúe una subrutina que calcule el factorial de un número.

La solución se encuentra en el archivo adjuntado.