

S3. PFA: Programar un pseudocódigo

Sitio: [Agencia de Habilidades para el Futuro](#)
Curso: Metodología de Pruebas de Sistemas 2º D
Libro: S3. PFA: Programar un pseudocódigo

Imprimido por: Eduardo Moreno
Día: martes, 26 de agosto de 2025, 00:47

Descripción



Inicio



Apertura



Desarrollo



Prácticas



Cierre

Tabla de contenidos

S3. Programar un pseudocódigo

Presentación de dominios

- Dominio 1: Factura digital
- Dominio 2: Taxi mobile
- Dominio 3: Sistema de GPS

S2. Diseñar casos de prueba

S1. Analizar un pseudocódigo

Introduccion a las Prácticas formativas

Práctica formativa	Objetivo	Semana
Autogestionada - individual ó entre pares- autocorrección	Construir una posible solución en pseudocódigo con las especificaciones del dominio elegido en la semana 2.	3

- **Elegí uno de los 3 dominios:** Factura digital - Taxi Mobile - Sistema de GPS
- **Recomendación:** Utilizar los conocimientos aprendidos en el 1er año de la carrera.
- **¿Dudas, consultas?:** En el caso de que lo necesites y requieras orientaciones del/la docente, [aquí](#) podrás dejar tus comentarios.

Compartimos orientaciones para desarrollar el pseudocódigo.

↓	1. Si lo hacés en grupo: Una reunión de planificación: Reúnanse como grupo y definan claramente el objetivo del software que van a construir, así como las especificaciones que se vieron la semana anterior.
↓	2. Análisis de requerimientos: Revisen detenidamente las especificaciones de la Historia de Usuario proporcionadas para entender qué funcionalidades y características debe tener el pseudocódigo.
↓	3. Diseño del software: Elaboren un diseño preliminar de la arquitectura del software, definiendo las diferentes partes del sistema y cómo se conectarán.
↓	4. Desarrollo del pseudocódigo: Comiencen a programar el pseudocódigo siguiendo las pautas de diseño previamente establecidas. Asegúrense de documentar adecuadamente el código y de seguir buenas prácticas de programación.
✓	5- Acceder a las claves de autocorrección: Les proporcionamos la claves de autocorrección, utilicenla para comparar sus resultados y verificar si sus casos de prueba están correctamente diseñados. Accedan desde aquí .

Es importante recordar que la colaboración y la comunicación efectiva dentro del grupo son clave para el éxito de este proyecto. Además, asegúrense de seguir las mejores prácticas de desarrollo de software y de cumplir con los plazos establecidos.

Práctica formativa	Presentación	Semana
Autogestionada e individual	¿Qué es un dominio?	2
<p>En el contexto de las metodologías de prueba de software, el término dominio se refiere al conjunto de valores y condiciones que son válidos y relevantes para un determinado aspecto o función del software que se está probando. En otras palabras, un dominio en este contexto representa el conjunto de todas las posibles entradas, salidas y condiciones bajo las cuales se espera que el software funcione correctamente.</p> <p>Cuando se realiza la prueba de software, es imposible probar todas las combinaciones posibles de entradas y condiciones debido a la complejidad y la cantidad de escenarios. Por lo tanto, los probadores de software seleccionan un conjunto representativo de valores y condiciones para evaluar el comportamiento del software. Este conjunto se conoce como el dominio de prueba.</p> <hr/> <p>El objetivo principal al definir y trabajar con un dominio de prueba es asegurarse de que se están cubriendo los casos más críticos y representativos del software, para garantizar que se haya evaluado adecuadamente su funcionalidad en diversas situaciones. La idea es identificar clases de equivalencia y particionar el espacio de entrada en conjuntos manejables para simplificar el proceso de prueba.</p> <p>En resumen, el concepto de dominio en metodologías de prueba se centra en la especificación y evaluación de conjuntos de datos y condiciones que son esenciales para la validez y la funcionalidad del software que se está probando.</p>		

Práctica formativa	Presentación	Semana
Autogestionada e individual	Dominio 1: Factura Digital	2



Una empresa que brinda el servicio de alarmas domiciliarias con monitoreo en todo el país quiere habilitar el servicio de envío de la factura por medios digitales a sus clientes adheridos al servicio, estableciendo una oportunidad para ahorrar recursos y mejorar el acceso y disponibilidad a la información.

Descripción de los requerimientos del cliente

Presentamos a continuación los requerimientos del cliente:

MPS TikTok Factura Digital



Mostrar transcripción del video

Historia de Usuario

La funcionalidad en particular que debemos probar se encuentra descrita en la Historia de Usuario que les compartimos a continuación:



En S2_PFA Dominio Factura digital encontrás la descripción del dominio que te presentamos antes en el video y la Historia de Usuario.

Práctica formativa	Presentación	Semana
Autogestionada e individual	Dominio 2: Taxi mobile	2



Esta empresa requiere desarrollar un sistema para celulares, para que los pasajeros de taxis puedan solicitar el taxi más cercano y saber su ubicación y demora en todo momento.

Descripción: Entrevista con el experto del dominio

Presentamos, a continuación, parte de la entrevista realizada al experto en el dominio:



Privacy policy

Equipo EAD · Dominio 2 Taxi Mobile

Mostrar transcripción del audio

Historia de Usuario

La funcionalidad en particular que debemos probar se encuentra descrita en la Historia de Usuario que compartimos a continuación.

En S2 PFA Dominio Taxi Mobile encontrarás la descripción del dominio que te presentamos antes en el podcast y la Historia de Usuario.



Esta empresa requiere desarrollar un sistema que permita a un conductor (entre otras funcionalidades), buscar un destino, obteniendo distintas alternativas para llegar hasta el punto marcado desde la ubicación actual.

Descripción: Entrevista al experto en el dominio

Presentamos, a continuación, parte de la entrevista realizada al experto en el dominio:

[Privacy policy](#)

Equipo EAD - Dominio 3 GPS

[Mostrar transcripción del audio](#)

Historia de Usuario

La funcionalidad en particular que debemos probar se encuentra descrita en la Historia de Usuario que compartimos a continuación.



En [S2_PFA Dominio Sistema GPS](#) encontrarás la descripción del dominio que te presentamos antes en el podcast y la Historia de Usuario.

Práctica formativa	Objetivo	Semana
Autogestionada - individual- autocorrección	Diseñar de casos de prueba con la estrategia de caja negra empleando dos métodos: partición de clases de equivalencia y pruebas de valor límite.	2

- **Elegí uno de los 3 dominios:** Factura digital - Taxi Mobile - Sistema de GPS
- **Herramienta:** : Tablas para confeccionar pruebas. [aquí](#)
- **Recordatorio:** Confeccionarás casos de prueba pero sin escribir código, porque solo estamos usando la estrategia de caja negra.
- **¿Dudas, consultas?:** En el caso de que lo necesites y requieras orientaciones del/la docente [aquí](#) podrás dejar tus comentarios.

Te compartimos orientaciones para diseñar los casos de prueba utilizando la metodología de caja negra para 1 (uno) de los 3 (tres) dominios es el siguiente:

↓	1- Leer los dominios y las historias de usuario: Comenzá leyendo detenidamente cada dominio y las tres historias de usuario para comprender qué funcionalidades debe tener el software y qué se espera que haga en cada caso. ¡Recordá que podés elegir uno solo para practicar!
↓	2- Releer para orientarte los ejemplos brindados: Volvé a revisar las historias de usuario en busca de ejemplos o casos específicos que te ayuden a entender mejor cómo debería funcionar el software en diferentes situaciones.
↓	3- Aplicar la metodología de caja negra: La metodología de caja negra se basa en evaluar el software sin conocer su estructura interna, centrándose en las entradas y salidas. En este caso, considerá las entradas que se pueden proporcionar al software y las salidas (los resultados) que se esperan obtener.
↓	4- Método 1: Identificar particiones de clases de equivalencia: Dividí las entradas en clases de equivalencia, es decir, agrupá valores de entrada que se espera que tengan un comportamiento similar. Por ejemplo, si el software acepta números, podrías dividirlos en clases de equivalencia como "números positivos," "números negativos" y "cero." Accedé a un ejemplos del método de equivalencias desde aquí .
↓	5- Método 2: Identificar pruebas de valor límite: Dentro de cada clase de equivalencia, identificá los valores límite o extremos. Estos son valores que podrían causar un comportamiento inusual o errores en el software. Por ejemplo, si estás probando un sistema de reserva de boletos de avión, los valores límite podrían incluir intentar reservar un boleto cuando no hay asientos disponibles. Accedé a un ejemplos del método de análisis de valores límite desde aquí .
↓	6- Usar las tablas de ejemplos de caso de uso: Utilizá las tablas de ejemplos de caso de uso que se proporcionan en el material de lectura como guía para organizar tus pruebas. Estas tablas te ayudarán a organizar las entradas, salidas esperadas y resultados reales de las pruebas.
↓	7- Diseñar los casos de prueba para cada historia de usuario: Para cada historia de usuario, creá casos de prueba siguiendo la metodología de caja negra. Asegurate de incluir las siguientes secciones en tus casos de prueba: <ul style="list-style-type: none"> - Descripción de la prueba. - Datos de entrada (valores de entrada). - Resultado esperado (qué debería producir el software como resultado). - Procedimiento de prueba (cómo llevar a cabo la prueba).

↓	8- Repetir el paso para otra historia de usuario que quieras usar.
📄	9- Acceder a las claves de autocorrección: Te proporcionamos la claves de autocorrección, utilízalas para comparar tus resultados y verificar si tus casos de prueba están correctamente diseñados. Hacé clic aquí .

Recordá que el enfoque de caja negra se centra en las entradas y salidas, por lo que no es necesario conocer la estructura interna del software. El objetivo es probar el software desde la perspectiva del usuario final y garantizar que funcione según lo esperado en una variedad de situaciones.

Práctica formativa	Objetivo	Semana
Autogestionada e individual	Analizar un pseudocódigo estáticamente para identificar variables y diseñar casos de prueba que verifique su funcionamiento.	1

- **Recomendaciones:** Es una práctica que retoma los conocimientos aprendidos en otras materias.
- **¿Dudas, consultas?:** En el caso de que lo necesites y requieras orientaciones del/la docente [aquí](#) podrás dejar tus comentarios.

El paso a paso para cumplir con la consigna de aplicar el análisis estático al código utilizando las tablas 1 y 2 se describe a continuación:

Tabla 1: Identificación de variables

Tabla 1 . Identificación de variables		
Tipo Variable	Tipo de dato	Nombre de la variable
Entrada/Salida		

- 1 Examiná el pseudocódigo proporcionado y comenzá por identificar todas las variables de entrada y salida que se utilizan en el programa. Estas variables son aquellas que el programa toma como entrada o produce como salida.
- **Tipo de variables:** Registrá el tipo de dato de cada variable, como entero, flotante, cadena, etc.
 - **Nombre de la variable:** Anotá el nombre de cada variable de entrada y salida.
 - **Entrada:** Indicá si la variable es una entrada (E) o una salida (S).

Tabla 2: Definición de casos de prueba

Tabla 2 - Identificación de casos de prueba				
Número de prueba	Entrada	Asignación	Operación	Salida

- 2 Después de identificar las variables de entrada y salida, procedé a definir tres posibles casos de prueba para el programa. Estos casos de prueba representarán diferentes situaciones que el programa deberá manejar.
- **Número de prueba:** Asigná un número de identificación único a cada caso de prueba para llevar un registro.
 - **Entrada:** Completá esta columna con valores específicos que se utilizarán como entradas para el programa en cada caso de prueba.
 - **Asignación:** Describí cualquier asignación o inicialización necesaria de las variables antes de ejecutar el programa.

- **Operación:** Especifica las operaciones o cálculos que se realizarán en el programa utilizando las variables de entrada.
- **Salida:** Indica el resultado esperado o la salida que se espera obtener después de ejecutar el programa con las entradas y operaciones definidas.

- 3 Repetí el proceso de llenar la tabla 2 tres veces, una para cada caso de prueba, asegurándote de que cada caso sea único y represente diferentes situaciones que el programa pueda enfrentar.
- 4 Una vez que hayas completado ambas tablas para el pseudocódigo proporcionado, tendrás un conjunto de casos de prueba y una descripción clara de las variables de entrada y salida. Estos casos de prueba pueden utilizarse para verificar el comportamiento del programa y detectar posibles problemas.

Este proceso te ayudará a realizar un análisis estático del código, identificar las variables relevantes y diseñar casos de prueba que cubran una variedad de situaciones posibles.

A continuación podrás descargar el archivo con los pseudocódigos para realizar la práctica formativa.

Clic para
descargar el
archivo



¿Qué son las prácticas formativas autogestionadas?

Al momento de dar los primeros pasos en una carrera estamos visualizando un futuro profesional y, en más de una oportunidad no tomamos conciencia que cada paso que damos forma parte de un ciclo de preparación, de formación, de desarrollo de habilidades y competencias que luego se pondrán al servicio de las diferentes relaciones socioculturales, económicas, productivas, etc. que caracterizan el mundo del trabajo.

Por lo que para adquirir aquellas capacidades es central poner el foco en procesos de aprendizaje centrados en prácticas formativas que nos permitan poner en acción y reflexión a partir de situaciones laborales reales o simuladas.

Así, en la materia "Metodología de pruebas de sistemas", pensamos prácticas formativas que promuevan las capacidades del perfil del/la egresado/a.

¿A qué nos referimos a que son "autogestionadas"?

- No son obligatorias, pero altamente recomendadas para poner en práctica lo contenidos de la materia y prepararte para los trabajos prácticos integrados que son calificados por el/la docente.
- Autocorrección y autoevaluación: Utilizá las claves de autocorrección para comparar tus soluciones con otras posibles respuestas. Esto te permitirá identificar tus áreas de fortaleza y aquellas en las que necesitás mejorar.
- Autonomía en el aprendizaje: Aprovechá la oportunidad de administrar tus tiempos y trabajar en estas actividades de forma autónoma. Establecé un plan de estudio y mantené una rutina para mantener un buen ritmo de trabajo.
- Explorá recursos adicionales: Si tenés dudas o querés profundizar en algún tema específico, buscá recursos adicionales, como libros, artículos o videos, para ampliar tus conocimientos.
- Interactuá con otros/as estudiantes: Trabajar con compañeros/as puede enriquecer tu aprendizaje. Podés formar grupos de estudio o participar en sesiones de estudio en línea para discutir los temas y resolver dudas de manera colaborativa.
- Preguntá al/la docente si es necesario: Aunque estas actividades no sean obligatorias, no dudes en escribirle al docente si tenés dudas o necesitás aclaraciones adicionales sobre las consignas. Ellos/as están allí para ayudarte en tu proceso de aprendizaje.