

Ejercicio Guiado 3

Enunciado



Desarrollo Dirigido por Pruebas

Desarrollo de Software

Marzo 2023

Grado en Ingeniería Informática - Doble Grado en Informática y ADE

ÍNDICE

OBJETIVOS	3
EXPOSICIÓN DE CASO	3
Requisitos Funcionales	4
Función 1. Solicitar el envío de un producto.	4
Entrada:	4
Proceso:	5
Salidas:	5
Interfaz	6
Función 2. Enviar un producto.	6
Entradas:	6
Proceso	7
Salidas	8
Interfaz	8
Función 3. Entregar Pedido	9
Entradas:	9
Proceso:	9
Interfaz	10
TAREAS A REALIZAR	12
NORMAS Y PROCEDIMIENTOS	13

OBJETIVOS

Este ejercicio guiado tiene tres objetivos principales:

- Practicar el proceso de desarrollo guiado por pruebas.
- Conocer las herramientas necesarias para el desarrollo basado en pruebas.
- Practicar las técnicas de pruebas funcionales y estructurales.

EXPOSICIÓN DE CASO

Los objetivos específicos de este ejercicio son definir, automatizar y ejecutar las pruebas unitarias requeridas que se utilizarán para verificar si el componente para la gestión de pedidos y productos funciona correctamente una vez que se complete su desarrollo.

Tienes una versión inicial del código fuente disponible en Aula Global. Ten en cuenta que este componente, en este momento, no implementa ninguna funcionalidad. El código se proporciona para que la implementación de los casos de prueba se pueda ejecutar utilizando la herramienta de automatización **PyBuilder**. Sin embargo, se espera que ninguno de los casos de prueba funcione correctamente ya que aún no se ha realizado el diseño y la implementación de las funcionalidades del componente.

Los estudiantes deben desarrollar las funciones propuestas basándose en el código fuente proporcionado. Deben aplicar las técnicas TDD junto con técnicas de prueba funcionales y estructurales.

Este componente debe ser desarrollado en Python 3.8, usando el IDE PyCharm y configurando un entorno virtual para el proyecto. El código debe compartirse en un repositorio privado en GitHub.

Requisitos Funcionales

Los requisitos funcionales para los que se desea definir los casos de prueba funcionales y estructurales son los siguientes:

Función 1. Solicitar el envío de un producto.

AM-FR-01: Para poder gestionar el envío de un producto a través del sistema primero hay que registrarlo. El sistema recibirá información del pedido y devolverá un código que será necesario para poder hacer su seguimiento. Este proceso es necesario para comprobar que los códigos de producto recibidos son correctos

Entrada:

AM-FR-01-I1: *product_id*. Es un código EAN13 que identifica el producto que debe ser servido.

AM-FR-01-I2: *order_type*. Puede tomar los valores “Regular” o “Premium”. Cuando el usuario es premium los pedidos serán tratados de forma especial posteriormente.

AM-FR-01-I3: *address*. Dirección a la que se envía el producto. (entre 20 y 100 caracteres con al menos 2 cadenas separadas por un espacio blanco)

AM-FR-01-I4: *phone_number*. Número de teléfono válido. (9 dígitos)

AM-FR-01-I5: *zip_code*. Código postal válido en España.

Proceso:

AM-FR-01-P1: El componente debe verificar la corrección de los datos recibidos.

AM-FR-01-P2: Si los datos recibidos son correctos, el componente obtendrá una firma mediante el algoritmo MD5. (Este valor MD5 se obtiene del método `__str__` de la clase `OrderRequest` (clase disponible en el código de Aula Global). Esta firma será el identificador del pedido y en adelante se denominará `OrderID`.

AM-FR-01-P3: El sistema debe almacenar en un fichero todos los datos de cada solicitud para poder comprobarlo posteriormente.

Salidas:

- AM-FR-01-O1: Una cadena en MD5 correspondiente identificador de pedido generado en formato hexadecimal.
- AM-FR-01-O2: Un fichero en el que se ha incluido los datos del pedido que se está registrando en el sistema.
- AM-FR-01-O3: Excepción en los siguientes casos:
 - El EAN13 recibido no es válido o no tiene un formato válido.
 - El tipo de pedido no es válido.
 - La cadena de dirección no es válida.
 - El formato del número de teléfono no es válido.
 - El código postal no tiene un valor válido.
 - Otros errores internos.

Interfaz

La definición del componente de interfaz que proporciona esta funcionalidad es la siguiente:

```
register_order (product_id,  
  
               order_type,  
  
               address,  
  
               phone_number,  
  
               zip_code):  
  
// Devuelve una cadena que representa AM-FR-01-01  
  
// En caso de errores, devuelve una OrderManagementException  
representa AM-FR-01-02
```

Función 2. Enviar un producto.

AM-FR-02: Para enviar el producto, el sistema debe verificar doblemente el identificador del pedido generado por la función anterior. Si el identificador del pedido es correcto el sistema generará un código de seguimiento que utilizará la empresa de transporte, con una fecha en que se espera entregar el producto.

Entradas:

AM-FR-02-I1: La ruta del archivo con los datos del producto que se va a enviar. El archivo de entrada JSON debe cumplir con el siguiente formato:

```
{  
  
  "OrderID": "<String having 32 hexadecimal characters>",  
  
  "ContactEmail": "<valid email>"  
  
}
```

Proceso

AM-FR-02-P1: el componente debe verificar que el pedido fue almacenado en el almacén de pedidos y que el OrderID coincide con los datos (es decir que no se han manipulado los datos del fichero).

AM-FR-02-P2: Si la petición es válida, el componente debe generar una instancia de la clase OrderShipping. Los atributos de esa clase son:

- "alg" (String). Identifica el algoritmo utilizado para firmar el código envío. Por ahora, su valor debe ser "SHA-256", pero en futuras versiones, el componente permitirá otros valores.
- "typ" (String). Por ahora, el componente solo permite el valor "UC3M" para este campo.
- "order_id" (String). Este valor es el valor del OrderID en AM-FR-01.
- "issued_at". Contiene la fecha en la que se envía el producto (hora UTC en formato de marca de tiempo). Será la fecha actual.
- "delivery_day". Fecha de entrega del producto (hora UTC en formato de marca de tiempo). Si el usuario que solicitó el producto es "Premium" esta fecha será el

día siguiente, mientras que en caso de “Regular” esta fecha será 7 días después.

- “tracking_code” (String). Este es el código que se podrá utilizar para seguir el envío durante su transporte. Este código se calcula codificando los campos anteriores usando “SHA-256”. El texto a codificar tiene el formato:

```
{alg:<value>,typ:<value>,order_id:<value>,issued_at:<value>,  
  delievery_day:<value>}
```

AM-FR-02-P2: El componente debe almacenar todos los envíos procesados correctamente en un fichero.

Salidas

AM-FR-02-O1: una cadena SHA-256 con la clave en formato hexadecimal, si los datos del producto que se han recibido son correctos (se ha encontrado en los pedidos almacenados y los datos están bien).

AM-FR-02-O2: Un fichero en el que se incluye los datos del envío procesado.

AM-FR-02-O3: OrderManagementException en los siguientes casos:

- No se encuentra el archivo de datos.
- El archivo no tiene formato JSON.
- El JSON no tiene la estructura esperada.
- Los datos del JSON no tienen valores válidos.
- El pedido no se encontró entre los pedidos registrados.
- Error de procesamiento interno al obtener la clave.

Interfaz

La definición del componente de interfaz que proporciona esta funcionalidad es la

siguiente:

```
send_product (input_file):  
  
// El archivo de entrada es una cadena con la ruta del archivo  
descrita en AM-FR-02-I1  
  
// Devuelve un String en hexadecimal que representa el código de  
seguimiento del envío (clave que será necesaria para  
AM-FR-02-O1  
  
// En caso de error, devuelve un OrderManagementException  
representa AM-FR-02-O2
```

Función 3. Entregar Pedido

AM-FR-03: Cuando el pedido llegue a su destino, el sistema registrará que se ha entregado al cliente. El sistema comprobará que el código de seguimiento es correcto y que el día es el programado (supondremos que sólo se puede entregar en la fecha prevista). Finalmente registrará la entrega en un fichero.

Entradas:

AM-FR-03-I1: Una cadena SHA256 que representa un código de seguimiento función 2 en formato hexadecimal.

Proceso:

AM-FR-03-P1: El componente debe verificar la exactitud del código de seguimiento recibido. En base a la experiencia de los métodos anteriores, debe determinar si la clave recibida es válida o no.

AM-FR-03-P2: Si el código de seguimiento es válido, el componente comprobará la fecha prevista de entrega.

AM-FR-03-P3: Si todo es correcto se registra la entrega en un archivo con la marca de tiempo (hora UTC) de la entrega y el código de seguimiento.

Salidas:

AM-FR-03-O1: El componente devolverá verdadero el código de seguimiento y la fecha son válidas. Se considera que la fecha de entrega es válida si la fecha es igual a la registrada.

AM-FR-03-O2: Un fichero con los datos de entrega registrados

AM-FR-03-O3: OrderManagementException en los siguientes casos:

- La cadena de entrada no contiene un código de seguimiento que pueda procesarse.
- El código de seguimiento no estaba registrado.
- El código de seguimiento no es válido (valores diferentes para cada situación)
- Error de procesamiento interno al procesar el código (es decir, archivos no encontrados, etc.)

Interfaz

La definición del componente de interfaz que proporciona esta funcionalidad es la siguiente:

```
deliver_product (tracking_number):
```

```
// la clave es una cadena con el valor descrito en AM-FR-03-I1
```

```
// Devuelve un valor booleano definido en AM-FR-03-O1
```

```
// En caso de errores, devuelve una OrderManagementException que  
representa AM-RF -03-02
```

TAREAS A REALIZAR

1. Cree un nuevo repositorio en GitHub para este ejercicio guiado. Nombra este repositorio de acuerdo con las reglas descritas en el ejercicio guiado 2 (GXX.2023.TYY.EG3) e invita a tu profesor de prácticas.
 2. Clona el proyecto e incluye el código y la estructura de carpetas disponibles en Aula Global.
 3. Siguiendo el proceso TDD, define los casos de prueba e implementa la **primera función**. Aplicar en este primer método el Análisis de Clases de Equivalencia y los Valores de Límite (cuando corresponda). Dependiendo de la naturaleza de los parámetros de entrada y salida, hay varias reglas a considerar. Estas reglas están relacionadas con:
 - Rango de valores de entrada / salida
 - Cantidad de valores de entrada
 - Semántica de los valores de entrada
 4. Después, define los casos de prueba e implementa la **segunda función**. En este caso, aplica la técnica de **Análisis sintáctico**. Si es necesario, **complementa** las pruebas **aplicando clases de equivalencia y valores límite**.
 5. Finalmente define los casos de prueba e implementa la tercera función. Aplicar en este tercer caso técnicas de **prueba estructural**.
 6. **El proyecto debe seguir el estilo de codificación PEP8**. Debe verificar el código **usando pylint con la configuración predeterminada**. Todos los errores y advertencias detectados por pylint deben resolverse.
 7. Ambos miembros del equipo deben interactuar con el repositorio de GitHub, siguiendo los principios de programación por pares y propiedad colectiva del código.
 8. Cree un documento PDF e incluya la definición de los casos de prueba para
-

cada función. Para la segunda función, incluye también la gramática y el árbol de derivación. Para la tercera función se incluye el Gráfico de control de flujo, la definición de las rutas básicas y los casos adicionales necesarios para probar los bucles. La buena presentación del documento se tendrá en cuenta en la nota (portada, encabezados, formato del documento, etc.)

Incluir también un fichero excel con la especificación de los casos de prueba descrita en el documento anterior (plantilla disponible en Aula Global).

Estos documentos deben incluirse en la carpeta "docs".

9. Generar un informe con los resultados de la ejecución de las pruebas para cada función. Incluir estos informes en la carpeta "docs\test_reports".

NORMAS Y PROCEDIMIENTOS

Este ejercicio se resolverá por parejas. Una vez finalizado el ejercicio, debes subirlo en el repositorio correspondiente de GitHub, incluyendo:

- El código fuente generado para cada funcionalidad durante el proceso TDD.
 - El código de las pruebas definidas para automatizar la validación de los casos de prueba seleccionados.
 - Los informes con los resultados de la ejecución de todas las pruebas, en una carpeta llamada "docs\test_reports".
 - Un documento PDF que contiene:
 - Las clases de equivalencia y el análisis de valores límite para la funcionalidad # 1.
 - La gramática, el árbol de derivación y la selección de casos de prueba a desarrollar para la funcionalidad # 2 utilizando la técnica de análisis sintáctico.
-

- El diagrama de flujo de control, las rutas básicas y los casos adicionales para probar los bucles para la funcionalidad # 3 usando la técnica de análisis estructural.
- Este documento debe guardarse en el directorio “docs” del proyecto.
- Un documento excel con la definición de los casos de prueba, de acuerdo con la plantilla disponible en aula global. Este documento debe guardarse en el directorio “docs” del proyecto.
- Un documento de word con la corrección cruzada de las pruebas del grupo que haya sido asignado.

Las pruebas del requisito funcional 1 se enviarán antes del 17 de marzo al grupo que le sea asignado. Los alumnos realizarán una corrección cruzada a sus compañeros de los casos de prueba identificados para esta función.

La práctica completa se entregará el 31 de marzo junto con un breve documento en el que se comente la evaluación de las pruebas del grupo que les haya correspondido.

De acuerdo con los estándares de evaluación continua establecidos en esta asignatura, si un equipo no publica la solución del ejercicio antes de la fecha límite, el ejercicio será evaluado con una puntuación de 0 puntos.