

ENUNCIADO

La empresa BK ha recibido un nuevo encargo de software.

Se trata de diseñar una aplicación para una tienda especializada en vender productos estéticos.

La tienda desea trabajar con software libre. Además, desea explícitamente que la aplicación sea capaz de cumplir las siguientes tareas:

- Proporcionar facturas de las ventas.
- Llevar la cuenta de lo que vende cada trabajador.
- Controlar el stock de productos en almacén.
- Operar con lector de código de barras y tarjetas de crédito.
- Controlar los precios de los productos y ofrecer la posibilidad de operar con ellos.
- El tiempo de respuesta de la aplicación ha de ser lo menor posible.
- No se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando simultáneamente.
- La empresa también quiere almacenar información de sus trabajadores: DNI, nombre, apellidos, número de la Seguridad Social, fecha de nacimiento, teléfono y localidad. Asimismo, de los productos interesa almacenar: código, marca, nombre comercial, precio, cantidad.

Tendrás que diseñar una planificación del proyecto de desarrollo de ese software que cumpla con las premisas estudiadas en la presente unidad de trabajo.

Esencialmente, el proyecto se divide en los siguientes apartados:

- A. Sintetiza el análisis de requerimientos del sistema para nuestro cliente. Plantea el diseño y determina el modelo de ciclo de vida más idóneo para esta aplicación.
 - B. Planifica la codificación, indicando el lenguaje de programación y las herramientas que usarías para la obtención del código fuente, objeto y ejecutable, explicando por qué eliges esas herramientas.
 - C. Planifica las restantes fases del ciclo de vida, indicando en cada una el objetivo que persigues y cómo lo harías.
 - D. Indica el ciclo de vida que usarías.
-

EJERCICIO

APARTADO A

La primera fase sería el análisis de requisitos que ha solicitado el cliente y que debe cumplir la aplicación correspondiente. Estos requisitos a su vez se pueden clasificar en **funcionales** y **no funcionales**. Los requisitos funcionales son aquellas funciones que tiene que realizar la aplicación considerando las respuestas que dará ante todas las entradas y cómo se comportará en situaciones inesperadas, en cambio, los requisitos no funcionales son los tiempos de respuesta del programa, la legislación que se aplica o el tratamiento ante varias peticiones. En nuestro caso los requisitos clasificados son:

Funcionales:

- Proporcionar facturas de venta
- Llevar la cuenta de lo que vende cada trabajador
- Controlar el stock de productos de almacén
- Operar con lector de código de barras y tarjeta de crédito
- Controlar los precios de los productos y ofrecer la posibilidad de operar con ellos
- Almacenar información del trabajador:
 - ◆ DNI
 - ◆ Nombre
 - ◆ Apellidos
 - ◆ Número de la SS
 - ◆ Fecha de nacimiento
 - ◆ Teléfono
 - ◆ Localidad
- Almacenar Información
 - ◆ Código
 - ◆ Marca
 - ◆ Nombre comercial
 - ◆ Precio
 - ◆ Cantidad

No Funcionales:

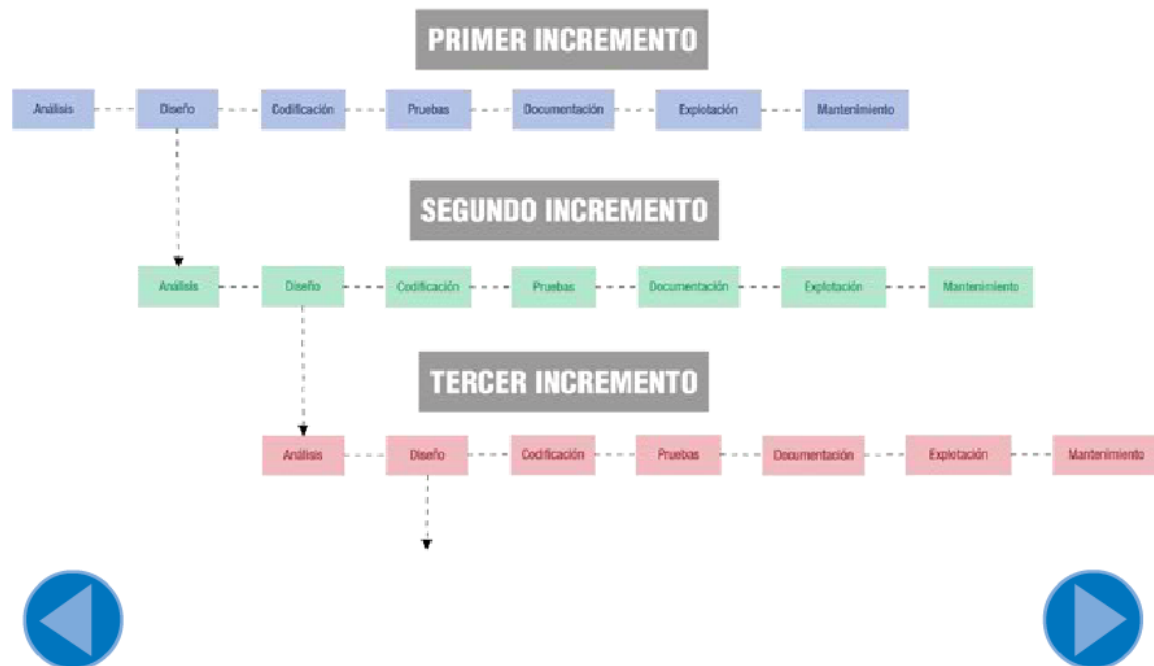
- El tiempo de respuesta de la aplicación ha de ser lo menor posible.
- No se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando simultáneamente.

DISEÑO

Tras analizar los requisitos del cliente pasaremos al diseño, donde escogeremos el modelo que se va a usar para desarrollar la aplicación. En este caso, el modelo idóneo para este proyecto sería el modelo iterativo incremental. Este modelo se caracteriza en la repetición

de cada fase, pero implementando una mejora de forma que tal mejora se propague en las siguientes fases. El motivo por el que se ha optado este modelo es en base a algunos requisitos solicitados como podría ser “operar con código de barras y tarjeta de crédito” porque gracias a este modelo si en un futuro desearan operar con lector de código de barras para realizar el inventario la aplicación debería ser capaz de realizar el primer requisito, además del segundo sin que dejase de funcionar lo que estaba hecho. Un esquema del modelo iterativo incremental sería el siguiente:

MODELO EVOLUTIVO: ITERATIVO INCREMENTAL



APARTADO B

HERRAMIENTAS

Como el cliente ha solicitado el deseo de usar programas de código abierto se han elegido los siguientes:

- Para el Desarrollo de la aplicación se usará el lenguaje **PYTHON**
- Como gestor de base de datos se usará **MySQL**

1. Obtención código fuente, código objeto y código ejecutable

Para facilitar la futura expansión de la empresa se optará por usar un lenguaje de programación multiplataforma como ya se mencionó en el apartado “Herramientas” de este documento.

Como sistema gestor de base de datos, se puede usar **MySQL** (Structured Query Language), puesto que se puede descargar de forma gratuita. Es un software potente, robusto y estable que cubre las necesidades expuestas por el cliente y sin aumentar los costes.

Para programar en Python nos ayudaremos del IDE [PyCharm](#), este nos ayudará a crear y gestionar la aplicación, así como a depurar el código de esta.

Finalmente Python interpreta el código objeto y lo ejecuta en la Máquina Virtual de Python, en donde podremos depurar dicho código hasta obtener su versión final o una que se acerque a esta.

APARTADO C

CICLO DE VIDA

2. Pruebas

La realización de pruebas es imprescindible para asegurar la validación y verificación del software elaborado. En esta etapa se prueban los programas para detectar errores y se depuran.

El fichero objeto se usa para realizar las pruebas unitarias mediante Unittest que es el entorno para pruebas de Python.

Una vez realizada con éxito las pruebas unitarias, realizaremos las pruebas de integración. Comprobaremos el funcionamiento de todo el sistema con todas sus partes interrelacionadas, para lo cual contactamos con el cliente con el fin de poder implementar en sus equipos una versión *Beta* que nos permita probar su desarrollo en el entorno donde se implantará definitivamente.

Dicho período de prueba será previamente pactado con el cliente.

3. Documentación

Todas las etapas en el desarrollo del software deben quedar perfectamente documentadas. Es necesario dar toda la información a los usuarios del software que hemos desarrollado y poder acometer futuras revisiones del proyecto.

Se debe ir documentando todas las fases del proyecto para poder pasar de una a otra de forma clara y definida. De esta forma crearemos los siguientes documentos:

- **Guía técnica:** El objetivo es facilitar un correcto desarrollo, realizar correcciones en los programas y permitir un mantenimiento futuro. Irá dirigido al personal técnico en informática.
- **Guía de uso:** Con este documento buscamos dar a los usuarios finales toda la información necesaria para utilizar la aplicación. Contendrá la descripción de la funcionalidad de la aplicación, la forma de comenzar a ejecutarla y ejemplos de uso. Este documento va dirigido a los usuarios que usarán la aplicación.
- **Guía de Instalación:** A través de este documento se pretende dar toda la información necesaria para garantizar que la implantación de la aplicación se realice de forma segura, confiable y precisa. Este documento va dirigido al personal informático responsable de la instalación en colaboración con los usuarios que la usarán.

4. Explotación y mantenimiento

Cuando a través de las pruebas nos aseguramos de que, el software funciona según lo previsto, carece de errores y las diferentes fases están documentadas, el siguiente paso es la explotación, que ocurre cuando se instala la aplicación en el equipo del cliente, se pone a punto y empieza a funcionar.

Una vez el software está en funcionamiento entramos en la etapa de mantenimiento, donde se realizan ajustes en el programa y se corrigen posibles errores que no han sido solventados durante la fase de desarrollo, para ello se hará uso de la **guía técnica**. Esta fase además incluye el lanzamiento de nuevas versiones del producto mejores que las anteriores que permitan al programa adaptarse a los cambios del entorno. Dentro del mantenimiento podemos encontrarnos con diferentes tipos de cambio como son:

- Cambios Perfectivos: Para mejorar la funcionalidad del software.
- Cambios Evolutivos: Nuevas funcionalidades, expansiones o eliminación del código en función de las nuevas necesidades del cliente.
- Cambios Adaptativos: Son las actualizaciones necesarias para adaptarse a las nuevas técnicas (o cambios) del mercado.
- Cambios correctivos: Corrección de errores/bugs que tenga el software en el futuro.

Todo esto será ofrecido al cliente en una tarifa la cual será establecida y acordada por ambas partes.