

ENUNCIAT (SOLUCIÓ)

L'empresa APPCHESTE ha rebut un nou encàrrec de programari.

Es tracta de dissenyar una aplicació per a una botiga especialitzada en vendre productes estètics.

La botiga desitja treballar amb **programari lliure**. A més, desitja explícitament que l'aplicació siga capaç de complir les següents tasques:

- Proporcionar factures de les vendes.
- Portar el compte del que ven cada treballador.
- Controlar l'estoc de productes en el magatzem.
- Operar amb lector de codi de barres i targetes de crèdit.
- Controlar els preus dels productes i oferir la possibilitat d'operar amb ells.
- El temps de resposta de l'aplicació ha de ser el menor possible.
- No es podran processar dues peticions alhora, encara que hi haja diversos equips funcionant simultàniament.
- L'empresa també vol emmagatzemar informació dels seus treballadors: **DNI, nom, cognoms, número de la Seguretat Social, data de naixement, telèfon i localitat**. Així mateix, dels productes interessa emmagatzemar: **codi, marca, nom comercial, preu, quantitat**.

Hauràs de dissenyar una planificació del projecte de desenvolupament d'aqueix programari que complisca amb les premisses estudiades en la present unitat de treball.

Essencialment, el projecte es divideix en els següents apartats:

- a) Sintetitza l'**anàlisi de requeriments** del sistema per al nostre client. Planteja el **disseny** i determina el **model de cicle de vida** més idoni per a aquesta aplicació.
- b) Planifica la **codificació**, indicant el **llenguatge de programació** i les **eines** que usaries per a l'obtenció del **codi font, objecte i executable**, explicant per què tries aqueixes eines.
- c) Planifica les **restants fases del cicle de vida**, indicant en cadascuna l'objectiu que persegueixes i com el faries.

SOLUCIÓN

Apartado a)

Análisis de requisitos

La primera fase sería el análisis de requisitos que ha solicitado el cliente y que debe cumplir la aplicación. 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 la aplicación 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

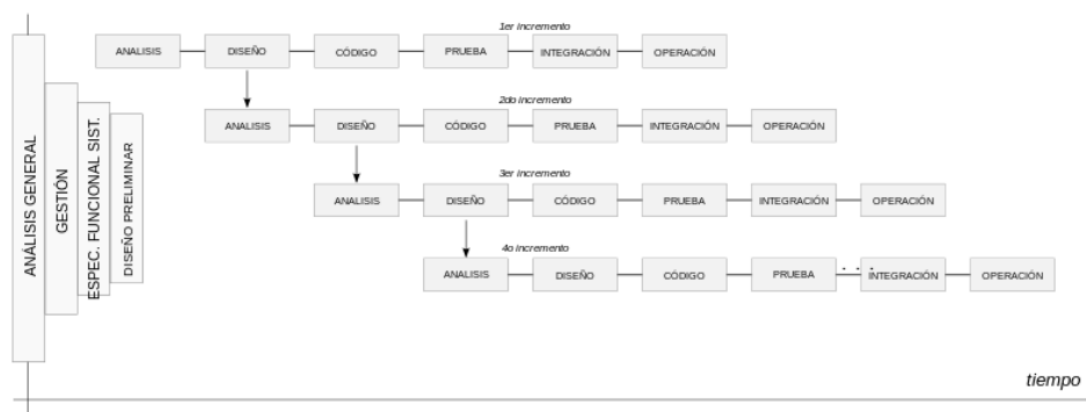
- Proporcionar facturas de venta
- Llevar la cuenta de lo que vende cada trabajador
- Controlar el stock de productos en almacén
- Operar con lector de código de barras y tarjeta de crédito
- Controlar los precios de los productos y ofrecer la posibilidad de operar con ellos
- Almacenar información del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

No funcionales

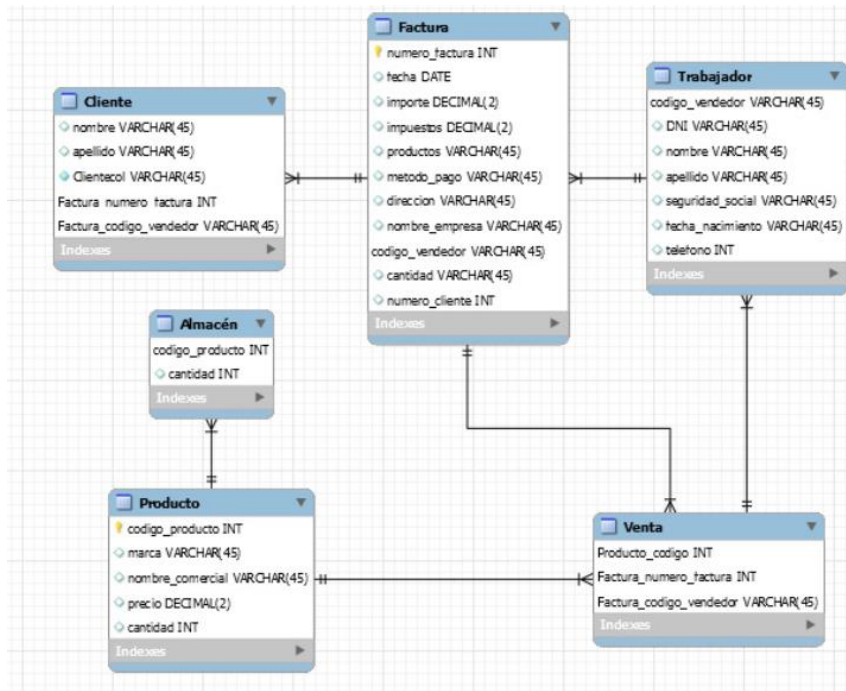
- El tiempo de respuesta de la aplicación ha de ser lo menor posible
- No se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando simultáneamente

Diseño

Tras analizar los requisitos del cliente pasaríamos al diseño, donde escogeríamos 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 esa mejora se vaya propagando a 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 lector de 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, es más como ya se conoce cómo trabajar con códigos de barras se habrá adquirido la experiencia necesaria que permita desarrollar la segunda función de forma más eficiente. Un esquema del modelo iterativo incremental sería el siguiente



Para facilitar el desarrollo del software se realizará un esquema para conocer cuáles son los principales elementos que lo componen y cómo estos interactúan entre ellos. Para la realización de este se ha usado el SGBD MySQL



Apartado b)

Herramientas

Como el cliente especifica que desea el uso de programas de código abierto se han elegido los siguientes

- Para el Desarrollo de la aplicación se usará el lenguaje JAVA
- Como gestor de base de datos se usará MySQL workbench

Obtención código fuente

Para programar en JAVA nos ayudaremos del IDE NetBeans. Para que este IDE sea capaz de ejecutar nuestro programa previamente necesitaremos instalar el Java Development Kit (JDK) el cual provee de un compilador y un entorno de ejecución conocido como Java Runtime Environment (JRE).

Obtención código objeto

Realizando la compilación del programa Java apoyándonos en NetBeans obtendremos el código Bytecode que se trata de instrucciones máquinas simplificadas específicas del lenguaje Java Obtención del código ejecutable

Tras la obtención del Bytecode este se interpreta y ejecuta en la Máquina Virtual Java (MVJ o Java Virtual Machine JVM).

Apartado c)

Las fases posteriores a la codificación serían:

Realización de pruebas

Una vez se ha obtenido el software una de las fases siguientes del ciclo de vida consisten en la realización de pruebas que nos permitirán comprobar de forma automatizada que el software cumple los requisitos demandados por el usuario. Las pruebas que se realizarán son:

- Pruebas unitarias. Al usar el lenguaje Java las pruebas unitarias se realizan con Junit y nos permitirán comprobar el correcto funcionamiento de las diferentes partes del programa.
- Pruebas de integración. Tras comprobar que las distintas partes que componen el software funcionan correctamente por separado se comprueba el sistema completo interrelacionando todas sus partes.
- Beta Test. Al superar las pruebas de integración se instalará el software en el equipo del cliente, de esta forma podemos comprobar el funcionamiento del programa en el entorno donde será usado por el cliente.

Documentación

Se elaborarán diferentes documentos para que el usuario conozca su funcionamiento y pueda resolver cualquier duda que le surja sin necesidad de contactarnos, ahorrando tiempo para ambas partes y, por otra parte, la creación de documentación permite facilitar futuras revisiones del software.

Las diferentes guías que se van a crear son las siguientes:

- Guía técnica. Queda constancia de cómo se diseñó la aplicación, la codificación del programa y las pruebas que se han realizado. Servirá como apoyo de nuestro equipo técnico para facilitar el mantenimiento del programa en el caso de que sea necesario realizar alguna corrección.
- Guía técnica. Reúne toda la información necesaria para poner en marcha la aplicación, explotarla. Está dirigida al personal informático que trabaja para el cliente y será el responsable de instalar la aplicación. Con ella se busca dar toda la información necesaria para que la instalación se realice correctamente.
- Guía de uso. Describe cómo funciona la aplicación, la forma de ejecutarla, incluye ejemplos de cómo usar el programa, los requerimientos de software para poder ejecutarla y la solución a posibles problemas que pueden ocurrir. Esta guía será

entregada al cliente para que conozca el funcionamiento de la aplicación y pueda solventar cualquier duda

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:

- Perfectivos. Para mejorar la funcionalidad del software.
- Evolutivos. Nuevas funcionalidades, expansiones o eliminación de código en función de las nuevas necesidades del cliente.
- Adaptativos. Son las modificaciones o actualizaciones necesarias para adaptarse a las nuevas técnicas del mercado
- Correctivos.