

1. Introducció

El desenvolupament de programes informàtics és una disciplina que integra diversos components del sistema informàtic i metodologies estructurades. Comprendre com interactuen aquests components i com els programes passen per diferents fases de desenvolupament és fonamental per a la creació de programari eficient i funcional. Un aspecte clau del desenvolupament de programari és la gestió de diferents tipus de codi: codi font, codi objecte i codi executable. Aquests conceptes, juntament amb altres com la generació de codi intermedi i l'ús de màquines virtuals, formen part de la base teòrica i pràctica que els desenvolupadors utilitzen diàriament.

2. Elements i eines implicades en el desenvolupament d'un programa informàtic

El sistema informàtic: memòria, processador i perifèrics

El desenvolupament de programari depèn de diversos components del sistema informàtic:

- **Memòria:** La memòria del sistema és essencial en el desenvolupament de programari perquè proporciona l'espai on s'emmagatzemen els programes en execució i les dades amb què operen. Hi ha diferents tipus de memòria, com la memòria RAM, que és volàtil i emmagatzema dades temporals, i la memòria ROM, que és no volàtil i emmagatzema dades permanents.
- **Processador:** El processador, també conegut com a CPU (Unitat Central de Processament), és el component del sistema informàtic que executa les instruccions del programa. La velocitat i eficàcia del processador són factors determinants per a l'execució ràpida i eficient del programari.
- **Perifèrics:** Els perifèrics són dispositius externs connectats a la unitat central de processament, com teclats, monitors, impressores i discos durs. Són vitals per a la interacció entre el programari i l'usuari, així com per a la gestió i emmagatzematge de dades.

El paper de les eines en el desenvolupament de programari

En el desenvolupament de programari, el programador utilitza diverses eines per facilitar la creació i execució del codi:

- **Compiladors:** El compilador converteix el codi font escrit pel programador en codi objecte, que és comprensible per la màquina però no encara executable. El procés de compilació pot identificar errors de sintaxi i estructurals que poden ocórrer durant la creació del codi.
- **Depuradors:** Els depuradors permeten als desenvolupadors identificar i corregir errors en el codi font. Aquesta eina permet executar el codi en passos controlats, observant els valors de les variables i la ruta d'execució, de manera que el programador pugui detectar on es produeixen els errors.
- **Entorns de desenvolupament integrats (IDE):** Els IDE són plataformes que integren diverses eines en una sola interfície. Aquests entorns faciliten la programació, la depuració, i fins i tot la compilació del codi, tot en un únic espai de treball.

3. Fases del desenvolupament d'un programa informàtic

Fases d'anàlisi, disseny, implementació i proves

Les fases en el desenvolupament d'un programa informàtic són fonamentals per garantir la qualitat i eficiència del programari creat. Aquestes fases inclouen:

- **Anàlisi:** En aquesta fase es recopilen els requisits del sistema i es defineixen les necessitats de l'usuari i les funcionalitats del programa. També es fa una avaluació de les possibles limitacions tècniques i dels recursos disponibles.
- **Disseny:** El disseny és la fase en què es defineix l'arquitectura del programari. En aquesta fase, es dissenyen els fluxos de dades, els algorismes, les interfícies d'usuari, així com la manera com interactuaran els diferents components del sistema.
- **Implementació:** Aquesta fase implica la creació del codi font que donarà vida al programa. Es fa servir un llenguatge de programació adequat i es generen els mòduls de codi que, més endavant, es combinaran en un programa complet.
- **Proves:** Un cop implementat el programa, es realitzen proves exhaustives per identificar qualsevol error o mal funcionament. Això pot incloure proves unitàries, proves d'integració i proves d'acceptació.

La importància de cada fase en el cicle de vida del programari

Cada fase és essencial per al desenvolupament d'un programa informàtic robust i fiable. La fase d'anàlisi defineix la visió general del programari, mentre que el disseny estableix els fonaments per a una implementació exitosa. Les proves són crucials per assegurar que el producte final compleixi els requisits i funcioni correctament.

4. Diferenciació dels conceptes de codi font, codi objecte i codi executable

Definició de codi font

El **codi font** és el conjunt d'instruccions escrites pel programador en un llenguatge de programació. Aquest codi és llegible per a les persones i descriu les operacions que el programa ha de realitzar. Els llenguatges de programació varien en sintaxi i estructures, però tots tenen un conjunt d'instruccions que poden ser traduïdes a codi objecte mitjançant un compilador.

El procés de compilació i codi objecte

El **compilador** converteix el codi font en codi objecte, que és una representació del codi en un llenguatge de màquina comprensible per la CPU. El codi objecte conté instruccions en un format que la màquina pot entendre, però encara necessita ser lligat amb altres parts del codi per crear un programa complet.

Codi executable i la seva relació amb l'execució en màquines físiques i virtuals

El **codi executable** és el resultat final de la compilació. És el codi que pot ser executat pel sistema operatiu. Aquest fitxer conté totes les instruccions necessàries per realitzar les operacions del programa en un entorn físic. En entorns de **màquines virtuals**, el codi pot ser adaptat perquè s'executi en un sistema virtual, proporcionant més flexibilitat i portabilitat.

5. Generació de codi intermedi i execució en màquines virtuals

Què és el codi intermedi

El **codi intermedi** és una forma de codi que no està completament optimitzat per a una plataforma específica, sino que es troba en una forma que pot ser interpretada per una màquina virtual. Aquest codi pot ser posteriorment compilat o interpretat per una màquina virtual, facilitant així l'execució de programes en diferents entorns.

L'ús de màquines virtuals en el desenvolupament d'aplicacions

Les **màquines virtuals** simulen un entorn de maquinari físic i permeten que el programari s'executi en diferents sistemes operatius o plataformes sense necessitat de modificar el codi original. Això és comú en llenguatges com Java, que utilitzen una màquina virtual per executar el codi intermedi generat per la compilació.

6. El cicle de vida del programari i les metodologies àgils

Introducció a les metodologies àgils

Les **metodologies àgils** es basen en un enfocament flexible i iteratiu per al desenvolupament de programari. En lloc de seguir un pla estricte des de l'inici, les metodologies àgils promouen la capacitat d'adaptar-se a canvis i feedback constant durant el cicle de vida del projecte. Els equips treballen en **iteracions curtes** per lliurar increments de programari que poden ser provats i millorat contínuament.

El cicle de vida de les aplicacions en el marc d'una metodologia àgil

En el marc de les metodologies àgils, el cicle de vida d'una aplicació és més flexible que en els models tradicionals. Les fases de desenvolupament es realitzen de manera iterativa, amb **sprints curts** per desenvolupar funcionalitats específiques i ajustar-se a les necessitats canviants del client.