Examen-2ndo-Cuatrimestre.pdf

RobertoMati

English for software developers

2º Grado en Tecnologías Interactivas

Escuela Politécnica Superior de Gandía
Universidad Politécnica de Valencia

PROGRAMMING LANGUAGES

- **WHAT IS THE MAIN DIFFERENCE BETWEEN THE IMPERATIVE AND THE DECLARATIVE PROGRAMMING PARADIGMS?**
  On the one hand, in the imperative programming paradigm, you can describe a sequential process of tasks, an algorithm, to solve a problem. In this type of paradigm, we focus on how the program operates. Python or C are examples of imperative programming languages.

  On the other hand, in the declarative programming paradigm, we focus on what the problem is and what the program should do to solve it, but not step by step. SQL is an example of a declarative programming language.

- **WHAT DO COMPILERS, INTERPRETERS AND ASSEMBLERS DO?**
  Assemblers converts/translates assembly language (low-level language) into machine code. It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor.

  Interpreters and compilers translate high-level language (more understandable for humans) into machine code. Interpreters parse the code instruction by instruction until the end of the program and shows an error as soon as it hits a problem, while Compilers read the source code as a whole and translates it in one go providing that it is free of errors.

- **DESCRIBE THE CLASSIFICATION OF PROGRAMMING LANGUAGES BY TYPING. GIVE SOME EXAMPLES.**
  Programming languages are classified according to "Type checking" or "Type safety".

  -Type checking depends on when the checking is done. We have "statically typed programming languages (PLs)", like C#, C or Java, where typing is checked prior to running the program (at compile time); and "dynamically typed PLs", like Python or JavaScript, where type checking happens after compilation (at runtime).

  -Type safety is related to the concept of type conversion; it prevents errors from happening. We have "weakly typed PLs", like JavaScript or C, where implicit type conversion is done, so type coercion is allowed; and strongly typed PLs", like Python or C#, where implicit conversion is prohibited because they want to ensure the correctness of the code, so it is explicit.

CONCURRENCY

- **WHAT IS THE DIFFERENCE BETWEEN CONCURRENT PROGRAMMING AND PARALLEL PROGRAMMING?**
  Concurrent programming is a sequential programming in which tasks are performed one by one, seeming to go at the same time but they do not. Just one task is being performed at a specific point of time. It is used in one core devices, where only one task can be done at once.

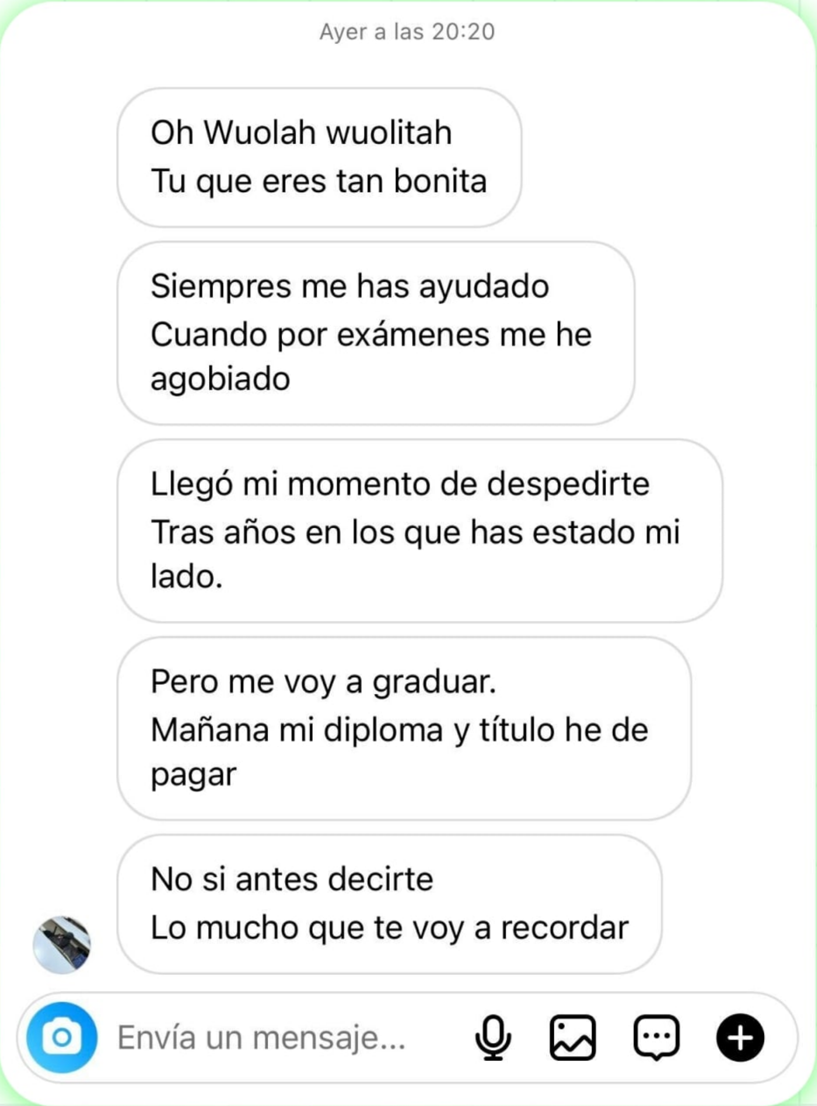
  Parallel programming is a simultaneous programming in which several tasks are being performed at the same time. It is used in multiple core devices and is more efficient than the concurrent one because more tasks are being performed at a specific point of time.

- **DESCRIBE THE TWO MAIN CONCURRENCY HAZARDS WHEN WRITING CODE: RACE CONDITION AND DEADLOCK.**
  Race condition occurs when two computer program processes, or threads, attempt to access the same resource or variable at the same time and cause problems in the system. They are a common issue for parallel programming.

  A deadlock is a situation in which two programs that are sharing the same resource or variable are preventing each other from accessing the variable, resulting in both programs ceasing to function.

Que no te escriban poemas de amor cuando terminen la carrera 🚩🚩🚩🚩🚩🚩

(a nosotros por suerte nos pasa) ☺



Ayer a las 20:20

Oh Wuolah wuolitah
Tu que eres tan bonita

Siempres me has ayudado
Cuando por exámenes me he agobiado

Llegó mi momento de despedirte
Tras años en los que has estado mi lado.

Pero me voy a graduar.
Mañana mi diploma y título he de pagar

No si antes decirte
Lo mucho que te voy a recordar

WUOLAH

SOFTWARE ENGINEERING

- **WHAT ARE THE TRADITIONAL STAGES IN THE SOFTWARE LIFE CYCLE?**
  There are usually five stages. The first stage is requirements analysis, where we start defining the project goals into defined functions and operations of the intended application and how the product will interact with the final user. We also define how do we start and what are we negotiating with the customer/client.

  The second stage is the Design stage, where we must create and think of solutions of the problems found in the previous stage.

  The third stage is the Development or Implementation stage, where you write the program. We must keep in mind modularity to make software more manageable.

  The fourth stage is the Testing stage, where we debug programs, so we focus on checking that the finish product is compatible with the software requirements.

  The final stage is Deployment, where we make the software run in the real environment for target users.

- **WHAT IS THE DIFFERENCE BETWEEN THE WATERFALL AND THE INCREMENTAL MODELS IN THE SOFTWARE ENGINEERING?**
  At the waterfall model, the stages are performed sequentially, and only when one stage is over, the next stage starts. It is easy to manage and good for small projects where requirements do not change, but there is no chance to change the requirements once they are agreed between both parties, and this can make the project become obsolete at the end.

  In the incremental model, the software involves multiple builds, and each build is a preliminary version of the product. This model generates software very quickly, testing is easier, and stakeholders can response to each build, so you can adapt to the changes of the requirements from the stakeholders.

- **WHAT IS DATA-COUPLING PROBLEM?**
  Data coupling is sharing data between modules, and it is a risk because modifications on one item of data in one module, can affect another module. It appears with global variables or when you pass data through parameters, so, to get rid of data coupling, we will have to maximize the independence among modules.

- **DESCRIBE THREE DIFFERENT TYPES OF SOFTWARE DOCUMENTATION: USER DOCUMENTATION, SYSTEM DOCUMENTATION AND TECHNICAL DOCUMENTATION.**
  User documentation describes the features of the software and how to use them. It is read by the user and depending on its quality, you can increase the sales, so it can serve as a marketing tool. It is included within the software.

  System documentation describes the internal composition of the software. It is maintained along the life cycle of the product. Here is where you use UML diagrams or follow conventions about how to write a program naming variables and so on.

  Technical documentation describes how the software should be installed and serviced.

- **WHAT IS THE DIFFERENCE BETWEEN BLACK-BOX AND WHITE-BOX TESTING?**
  The black-box methodology focuses on what the input data is and what the expected output should be. We focus on the interaction with the application and the results i.e., we check the user’s experience with the product. It is easy to use and very fast.
  In contrast, the white-box methodology focuses on the inner workings of the application, checking security holes, memory leaks, speed processing. In this kind of methodology, you perform unit testing for each block of code and every possible internal interaction is examined. It is used when you want to check that your program or application works as efficient as possible.

AGILE AND SCRUM

- **DESCRIBE FIVE PRINCIPLES IN THE AGILE MANIFESTO.**
  The first principle is that Business Priorities change. Changes are unavoidable, we should adapt to changes and be flexible. This is going to increase business value.

  The second principle is to deliver working software as often as possible because this is a measure of progress.

  The third principle is to work closely with the domain experts because they know why the software works in that way. They will use the software, so they have experience using the software.

  The fourth principle is that do not bother to write a lot of documentation before developing the system but create the documentation when it is needed.

  The fifth principle is that it is necessary to collaborate between developers and non-developers, i.e., all the team.

- **DESCRIBE THE FOLLOWING TWO ARTIFACTS IN SCRUM: PRODUCT BACKLOG AND SPRINT BACKLOG.**
  Product backlog is a list of all work that remains on a project, a list of prioritized tasks to be completed by the team, like a To-Do list. It is going to reflect customers' needs. The Product Owner manages the product backlog, he can add and remove user stories (concise description of a feature that the user wants the product to have).

  Sprint backlog is a list of all work that remains to be done by the team in a sprint. It is a subset of the product backlog. The user stories are divided into tasks that the members of the team must do during the sprint. At the end of the sprint, the sprint backlog must be empty.

- **DESCRIBE THE FOLLOWING THREE ROLES IN SCRUM: SCRUMMASTER, PRODUCT OWNER AND DELIVERY TEAM.**
  The Scrum Master is a servant leader, the team coach. He/She looks after the team's interest and ensures that there are no obstacles for the team. He/She should be a good communicator and a conflict manager.

  The product owner represents a costumer, looks after the customer's interest, and meets with the customer to determine their needs and prioritize those items. He/she manages the product backlog. His/her role is more important before and after the sprint rather than during the sprint.

  The delivery team consists of programmers, testers, front-end designers, etc. The work in the sprint as a self-organizing unit. Each member knows all aspects of the product, but each is an expert on a few aspects.

- **DESCRIBE THE FOLLOWING THREE ACTIVITIES IN SCRUM: SPRINT PLANNING, DAILY STAND-UP, AND SPRINT REVIEW.**
  The sprint planning occurs before the sprint, in this activity we decide the tasks of the sprint and the length of each task. The delivery team, scrum master and product owner participate in this activity.

  Daily stand-ups occur during the sprint once daily, no more than fifteen minutes and they talk about how their work is progressing or their problems. The delivery team, Scrum master and product owner are involved.

  Sprint reviews happen at the end of the sprint, where the delivery team presents the user stories that have been completed during the sprint to the customers, who can give feedback of the product demo, the scrum master, and the product owner.

GRAPHS

- **DESCRIBE THE FOLLOWING TERMS RELATED TO GRAPHS: DEGREE, ORDER, SIZE, WEIGHT, AND PATH.**

  The degree of a vertex of a graph is the number of edges that are incident to the vertex. There are in-degrees, which are the number of edges coming into a vertex in a directed graph, and out-degrees, which are the number of edges going out of a vertex in a directed graph.

  The order of a graph is the number of vertices in the graph. The size of a graph is the number of edges in the graph.

  The weight is the sum of edges a graph has. The path is in a graph is a succession of adjacent edges, with no repeated edges, which joins two vertices.

- **DESCRIBE THE FOLLOWING TERMS RELATED TO GRAPHS: GEODESIC PATH, DIAMETER, LOOP, AND CYCLE.**

  A geodesic path between two nodes in a graph is a path with the minimum number of edges. If the graph is weighted, it is a path with the minimum sum of edge weights.

  The diameter of a graph is the length of the longest geodesic path.

  A loop is an edge that connects a vertex to itself.

  The cycle is a non-empty trail in which only the first and last vertices are equal.

- **HOW DOES DIJKSTRA'S ALGORITHM WORK?**

  This algorithm finds the shortest path between a given node and all other nodes in a graph. It uses the weights of the edges to find the path that minimizes the total distance (weight) between the source node and all other nodes.

  To do this, we follow the next steps: First, we focus on the vertex with the smallest cost. Then, we check the neighboring nodes and calculate the distance for the neighboring nodes by summing the cost of the edges leading from the start vertex. Finally, if the distance (cost) to a vertex we are checking, is less than a known distance, we update the shortest distance for that vertex.

MALWARE

- **WHAT IS THE DIFFERENCE BETWEEN BLACK, WHITE AND GREY HATS IN COMPUTING SECURITY?**
  Black hats, also known as cracker, are an illegal type of hacker that searches vulnerabilities in the software to steal data, destroy data or to gain financial benefits. They access data without permission, so they are acting illegally.

  White hats use knowledge for good rather than evil. They are contracted as security specialists to find security holes of the company using the same methods that black hats do, but they are acting legally because they have the permission of the owner/company.

  Grey hats look for vulnerabilities without owner's permission. If security issues are found, they will report them to the owner and sometimes request a small fee to fix the issue. The intention is not malicious, but the activity is illegal.

- **DESCRIBE THE CHARACTERISTICS OF INFORMATION-SECURITY SYSTEMS.**
  Confidentiality refers to the fact that information is restricted to authorize people or devices.

  Integrity refers to the fact that information should be reliable, trustworthy. There are two types; data integrity, the system checks that data has not been altered, and owner integrity, the system checks that the owner is correctly identified (authenticity).

  Availability refers to the fact that information should be available to users.

- **DESCRIBE TELEPHONE SCAM, HOAXING AND PHISHING AS SOCIAL-ENGINEERING ATTACKS.**
  In social engineering, the hacker uses psychological manipulation to trick the user through social contact, and these are a few examples.

  Telephone scam is a form of fraudulent activity with the goal of stealing money or information impersonating someone or a company.

  Hoaxing is a message warning a user for a fake malicious thing and wants them to solve it by doing something. The message is usually a chain e-mail.

  Phishing consists in stealing users' data, like login credentials or credit card numbers, masquerading as a trusted entity. They dupe the user into opening an email, a website, a message, etc.

- **DESCRIBE THE FOLLOWING TYPES OF NETWORK ATACKS: SNIFFING, SPOOFING, RANSOMWARE AND DDOS.**
  Sniffing, also known as eavesdropping, is a process of monitoring and capturing data packets passing through a network. Network administrators use sniffers to monitor and troubleshoot network traffic and used by attackers to capture data packets containing sensitive information.

  Spoofing involves a cybercriminal masquerading as a trusted entity or device to get you to do something beneficial to the hacker but detrimental to you.

  A Ransomware is a malware that prevents or limits the user from accessing the system, either by locking the system's screen or locking the user's files for money.

  DDOS, Distributed Denial of Services, is a type of DoS that involves multiple connected online devices which are used to saturate websites with fake traffic.

- **COMPARE HOW VIRUSES, TROJANS AND WORMS WORK.**

  A virus is a computer program that can copy itself and infect a computer without permission or knowledge of the user. It disturbs the computer behavior and spreads by replication although it is downloaded by humans.

  A trojan is a type of malicious code or software that looks legitimate but can take control of the computer. It is designed to damage, disrupt, or steal information. It spreads by human intervention because it needs to be installed or downloaded.

  A worm is a standalone program, with a huge capability to travel across networks. Its aim is self-replicate and infect other computers while remaining active on infected systems. It duplicated itself to spread to uninfected computers.

ENCRYPTION, AUTHENTICATION AND AUTHORIZATION

- **DESCRIBE THE “CAESAR CIPHER” FORMULA, E.G WHEN ENCRYPTING THE PLAINTEXT “SECRET” AND THE KEY IS 10.**
  A Caesar Cipher is a method of encoding messages. Its formula consists in a substitution method where letters in the alphabet are shifted by some fixed number of spaces to yield an encoding alphabet. In this example, (using the English alphabet, 26 letters, from 0 to 25), the way to do it is as follows:

  In the case of letter S, we have to get its value when replacing it for a number, which is 18. Now, as the given key is 10, we have to count ten positions starting from 18, and as in the English alphabet the maximum number is 25, when we reach that number, we have to keep counting from 0. In this case the result number is 2, which is from letter C.

  If we follow this method with each letter of SECRET, we get that the result of the encryption is COMBOD.

- **HOW DO SYMMETRIC-KEY AND ASSYMETRIC-KEY SYSTEM WORK?**
  Symmetric and asymmetric systems are used to encrypt or decrypt information between two devices.
  The symmetric encryption uses the same key to encrypt and decrypt data making it quite easy to use. In asymmetric encryption, a public key is used to encrypt data and a private key is used to decrypt information. This type of encryption is safer and avoids attackers.

- **COMPARE ADVANTAGES AND DISADVANTAGES BETWEEN SYMMETRIC-KEY AND PUBLIC-KEY ENCRYPTIONS (E.G, SECURITY, SPEED AND NUMBER OF KEYS).**
  The symmetric-key encryption is faster but less secure than asymmetric-key encryption. Symmetric encryption only uses one key to encrypt and decrypt, while asymmetric encryption uses one public key to encrypt and a private key to decrypt, this explains why asymmetric encryption is slower but safer.

- **EXPLAIN HOW SSL ESTABLISHES ENCRYPTED COMMUNICATION BETWEEN A SERVER AND A CLIENT.**
  SSL establishes encrypted communication following the next steps:

  First, the web browser connects to the SSL secured web server. Next, the web server that contains the SSL certificate plus a private key, sends a copy of the SSL certificate to the web browser. Then, the web browser checks that the certificate authority is trustworthy.

  If it is trustworthy, then, the web browser uses the public key in the SSL certificate to encrypt symmetric key and sends this encrypted symmetric key through the URL.

  Consequently, the web server uses the private key to decrypt the symmetric key, so the web server gets the symmetric key that has been sent by the web browser. Now, this symmetric key is going to be used as the session key.

  Finally, every requested html document (web page) is going to be encrypted with the symmetric key, so now the web server and the web browser encrypt all transmitted data with the key.

- **DESCRIBE THE MOST IMPORTANT BIOMETRIC TECHNIQUES.**
  Face recognition involves analyzing features that are common to everybody’s face, like the chin, the jaw lines, the shape of the mouth, etc. It does not need the collaboration of the subject, which is good, but distinctiveness is no guaranteed, it does not work properly with poor light conditions, it is affected by variation in facial expressions; so, it’s not very reliable.

Fingerprint recognition is more reliable than face recognition, but it requires physical contact, and if there is some damage on the fingerprint, it won’t work properly. It is a cheap technique.

Iris recognition is a reliable, accurate and distinctive feature. Diseases can alter the iris and high-quality images can deceive the scanner. It is expensive.

Palm recognition captures a vein patron image of your palm. It is very secure and accurate, but it is expensive.

Voice recognition is not very reliable, has a high rate of errors, and can be affected by the noise. It is easy to spoof.

IOT

- **GIVE A DEFINITION OF IOT AND SOME EXAMPLES FOR MEDICAL APPLICATIONS.**
  IoT consists in interconnecting devices on the internet to exchange data from observations and to make it available to anyone. Its goal is to produce, consume or present data about some event. Some examples for medical applications are the connection of devices that control health like pacemakers or glucose monitors.

- **GIVE A DEFINITION OF IOT AND SOME EXAMPLES IN THE FIELD OF THE VEHICLE INDUSTRY.**
  IoT consists in interconnecting devices on the internet to exchange data from observations and to make it available to anyone. Its goal is to produce, consume or present data about some event. In the vehicle industry, IoT can be used to generate maintenance reports that are sent automatically via email or by phone, to control the state of your car through your mobile phone, etc.

- **WHAT IS THE RASPBERRY PI? DESCRIBE ITS MAIN COMPONENTS.**
  The Raspberry Pi is a board that can be used in projects that need the capability of a computer with the real operating system but with the advantage of a reduced size.

  It has several components like a HDMI connector, to connect the Raspberry Pi to a monitor (not compulsory), a CPU, a microprocessor, RAM memory, USB Ports, for keyboards and mouse, for example, an Ethernet port to connect the Raspberry to the internet, a camera connector (CSI), a display connector (DSI), to connect, for example, a camera for remote video monitoring, a Micro SD card slot to store data and the operating system and GPIO pins, used to connect electronic devices, like sensors.

- **DESCRIBE THE MAIN DIFFERENCES BETWEEN RASPBERRY PI AND ARDUINO.**
  The main difference between them is that Arduino is a microcontroller board, while Raspberry Pi is a microprocessor-based minicomputer (SCB).

  The Microcontroller on the Arduino board contains the CPU, RAM, and ROM. All the additional hardware on Arduino Board is for power supply, programming, and IO connectivity. Raspberry PI SCB has all features of a computer with a processor, memory, storage, up to 256GB thanks to the micro-SD slot, graphics, driver, connectors on the board.

  Raspberry Pi needs an Operating System to run, and you can install some applications, while Arduino just needs a binary of the compiled source code.

  Arduino needs an extra piece of hardware to provide connectivity, while Raspberry Pi doesn’t.

- **DESCRIBE THE DIFFERENCE BETWEEN ACTIVE AND PASSIVE SENSORS AND GIVE SOME EXAMPLES.**
  Active sensors are sensors with an energy source inside. They send out radiation and measure that backscatter reflected to it. An example of an active sensor is a Radar or a GPS.

  Passive sensors detect radiation that are emitted by other objects, but not its own. A common example is the thermometer.