1. **Descrever um plano genérico cobrindo, pelo menos, a parte da especificação do sistema (não da app), o desenvolvimento, testes e aceitação/instalação.**

   1) Training and education - Identify all possible technologies, applicable standards and laws; Provide training on these technologies (e.g. bluetooth, wifi, energy charging...), on the applicable standards and on the relevant laws to a specific amount of engineers that will be involved in the project (not all engineers might need all trainings)

   2) Project Planning - Define a high level schedule (with the major project phases and milestones); Include these 6 phases on that plan (not necessarily divided as such); Plan for standards, tools, metrics requirements. Appoint a project responsible for management and define the communication interfaces with the customer; Perform project staffing, management, risk management, etc...

   3) Threat Analysis / Security Requirements – Perform a complete threat analysis, by gathering all information about the planned technologies and tools, by analysis information from similar projects and by collecting the most common vulnerabilities applicable to the technologies to be used. From the threat analysis derive security requirements to cope with the identified threats.

   4) Design and Architecture – Functional requirements – Complete the functional requirements gathering, get requirements approval from the customer. Write down an architecture and derive a specific design for the system. Get also the customer approval on the architecture and design (if necessary do a demo or prototype).

   5) Implementation – Create two teams: one to focus on the development path and one to test the system developed. Dev path: implement the solution, perform source code analysis, and unit test it, correct issues found and received from testing team and issue new versions of the system. Test path: Integrate the solution with a simulated environment, develop a test strategy (penetration testing, fuzzing, functional testing, etc), test specs to cover all the functional requirements and the security requirements. Report the identified issues. Test the same solution in the target real environment (might need test strategy adaptation). Produce test reports.

   6) Support the deployment - Establish an incident response plan and a vulnerability disclosure system. Provide updated and tested version of the system.

2. **À luz de uma análise de ameaças de segurança (threats analysis) aplicável a este sistema:**

• Identificar os principais atributos de segurança que se devem ter em conta, e explicar brevemente cada um deles com alguns exemplos (3 atributos no mínimo, 6, no máximo).

You can/could choose between Confidentiality, Integrity, Non-repudiation, Authenticity, Accountability and Compliance.

Confidentiality – the solution will store personal and confidential data that needs to be protected by the appropriate means.

Integrity – We can have data integrity issues (for example data store in the server, such as personal data, history or payments, integrity of data transmission and, integrity of the application running on both the serve side or the local device (that can be protected for example by a checksum at startup).

Non-Repudiation – probably not very important for this case, but can be applicable if criminal investigation is required (see requirement example in next question).

Authenticity – An essential attribute, o log in, manage account information, start charging, etc.

Accountability – this is relevant for the fact that one's account is being changed (or one's credit card), so, charging under someone else's account without their permission should not be allowed.

Compliance – Compliance laws and rules (e.g. GDPR) must be followed for the storage and transmission of user data. It might even be possible that some of the data asked in the description of the solution might not be possible to be stored in the servers…

3. **Escrever entre 5 e 10 requisitos relacionados com a segurança funcional do sistema, cobrindo tópicos relacionados com os atributos apresentados na questão 2. Fazer esse mapeamento, mesmo que alguns requisitos mapeiem com mais do que um atributo.**

Alguns exemplos:

1. The system shall only allow access with a valid user id and password. (confidentiality, authenticity)
2. The user shall be prompted to define a 6 digits pin to unlock the terminal screen during the first login. (confidentiality, authenticity)
3. After a specified inactivity time, the system shall lock the screen. (confidentiality, authenticity)
4. Unlocking the screen shall be done by successfully entering the 6-digits pin. (confidentiality, authenticity)
5. After three invalid entries of the unlocking pin the system shall log the user out. (confidentiality, authenticity)
6. The data transmission with the server shall be packed and encrypted. (integrity, confidentiality)
7. The terminal software shall log all user operations. (Non-repudiation, Accountability)
8. The terminal, upon startup, shall verify the checksum of the application (to be compared with a hardcoded/unchangeable value). (Integrity)

9. The wifi and Bluetooth of the terminal shall be hidden. (Integrity)
10. The terminal shall not display the full phone number and credit card number of the user. (Compliance)

4. Definir um conjunto de testes para confirmar a correcta implementação dos requisitos escritos na questão 3. Mapear cada caso de teste ao(s) respectivo(s) requisito(s).

Alguns exemplos:

1. The system shall only allow access with a valid user id and password. (confidentiality, authenticity): Test 1: Perform tests of login with valid and invalid credentials, empty login, empty passwords, etc.
2. The user shall be prompted to define a 6 digits pin to unlock the terminal screen during the first login. (confidentiality, authenticity): Test 2: Create a new user, upon first login confirm that the application requests to set a pin. Test the pin with 6 digits, try also to define a pin with more and less digits (should fail). Logout that user and log back in, no prompt to define the pin should be shown now.
3. After a specified inactivity time, the system shall lock the screen. (confidentiality, authenticity): Test 3: Log in into the system. After the defined inactivity time (e.g. 30 seconds) very that the screen is locked.
4. Unlocking the screen shall be done by successfully entering the 6-digits pin. (confidentiality, authenticity): Test 4: Once the screen is locked try to unlock it without pin, with the correct pin but without the last digit, with and invalid pin 8all these situations should fail). Unlock the system with the correct and define 6 digits pin.
5. After three invalid entries of the unlocking pin the system shall log the user out. (confidentiality, authenticity). Test 5: Log in into the system. Wait for the screen to lock. Enter a wrong pin 3 times. Check that the system logs the user out. Try also to unlock the screen with 2 invalid pin entries and a correct third pin entry (this should work).
6. The data transmission with the server shall be packed and encrypted. (integrity, confidentiality): Test 6: With a sniffing tool capture several different types of communications with the server (e.g. password change, pin set-up, user data consultation, user data updates). Confirm that in all cases the data is encrypted and unreadable.
7. The terminal software shall log all user operations. (Non-repudiation, Accountability): Test 7: Once all the previous tests have been performed collect the terminal logs (by the maintenance interface). Confirm that all the operations performed in the previous tests have been properly logged. (Note: the requirements do not specify it but this test should also confirm that no confidential data is stored in the logs).
8. The terminal, upon startup, shall verify the checksum of the application (to be compared with a hardcoded/unchangeable value). (Integrity): Test 8: To test this a modification might be needed. Either modify some application location that does not affect the functioning of the application, or, preferably, slightly modify the application (e.g. add one useless line of code) and do not change the hardcoded value. If this value is instead stored in memory, then the test can simply be done by changing the

value and restarting the application. Confirm that the application detects the problem. Eventually confirm that the problem has also been logged.

9. The wifi and Bluetooth of the terminal shall be hidden. (Integrity): Test 9: With a laptop and network analysis tools try to capture the wifi and Bluetooth network ids close to the terminal. No id should be visible.
10. The terminal shall not display the full phone number and credit card number of the user. (Compliance): Test 10: Log in into the application. Do a consultation of the user data. Confirm that that the phone number and credit card number are not fully visible, for example, only the first 2 and the last digit are visible, the rest of the digits are replaced by "*". Try changing the data, confirm that when typing the phone number and the credit card number the same behaviour is observed. Collect the terminal logs and confirm that both phone number and credit card number are not stored in the log files.

5. Para testar a robustez da solução, fornecer os seguintes dados:

• Lista de todas as interfaces internas e externas da solução.

• Nomear e descrever (até 100 palavras cada) dois ataques de segurança que poderiam ser descobertos ao aplicar uma metodologia de penetration testing.

• Apresentar 3 soluções para evitar ataques de DOS (Denial-of-Service).

External interfaces: Wifi network, Bluetooth network, touch screen, network to communicate with the server (can be cabled or 5G, for example), maintenance local interface (e.g. USB port).

Internal interfaces: Interface with the storage disk, interface with the electricity system for charging the vehicles.

With penetration testing (automated or not) we can consider for example user enumeration attacks, to try to identify valid logins. Password guessing is the next step (with penetration testing and by using, for example brute force or list of common passwords).

To avoid or limit DOS attacks the system must be designed in order to protect the existing interfaces in order to avoid service disruption. The first, and most obvious DOS attack is a physical one by damaging or destroying the terminal. Disrupting the terminal power supply might also be considered a DOS. If the malicious actor can access any of the terminal interfaces (e.g. by being a valid user and using the mobile app, or by being able to sniff and interfere in the communication link between the terminal and the server. In order to avoid DOS attacks the system shall be detect and block any attempts. For physical attacks the terminal shall be designed to be robust and to avoid having the power supply lines visible. For interfaces attacks the software solution shall be able to detect traffic anomalies and enter into a secure more, for example reject the packets, drop the on-going session, etc.

6. O cliente identificou 4 ameaças que pretende ver resolvidas com a própria solução (sem intervenções externas, nem interrupções no serviço), e estas são:

• A) Instalação de vírus ou malware através do terminal

• B) Acesso indevido a funções de configuração do terminal (deve ser permitido apenas com o utilizador "maintenance")

• C) Obtenção de dados confidenciais de outro utilizador

• **D) Utilização de uma conta alheia para carregamento**

No âmbito de uma análise de ameaças (hazard analysis) fornecer, para cada ameaça listada pelo cliente, a seguinte informação:

- Possíveis consequências;
- Possíveis causas;
- Algumas medidas a implementar/adoptar para eliminar a ameaça.

A)

- Possíveis consequências: capture of user data, interruption of service, damage caused to the terminal or to the vehicles.
- Possíveis causas: Non authorized access to the maintenance interface (or malicious authorized access)
- Algumas medidas a implementar/adoptar para eliminar a ameaça: Perform checksum of the program memory, protect the maintenance access, avoid any user from uploading any file to the terminal.

B)

- Possíveis consequências: capture of user data, interruption of service.
- Possíveis causas: stolen of detected password (in this case the user id is known)
- Algumas medidas a implementar/adoptar para eliminar a ameaça: protect the maintenance access (e.g. multi-factor authentication).

C)

- Possíveis consequências: obteenção de dados confidenciais, cartão de crédito, morada, telefone para efectuar outros ataques.
- Possíveis causas: descuido do utilizador, accesso indevido ao terminal.
- Algumas medidas a implementar/adoptar para eliminar a ameaça: Não guardar nenhuma informação confidencial no terminal, não afixar nenhuma informação confidencial no terminal (apenas parte), encriptar todas as transacções, não loggar info confidencial.

D)

- Possíveis consequências: Furto, utilização indevida ao carregar uma viatura e forçar o pagamento a outro utilizador, utilizador fica insatisfeito e entr com processo contra a emprea de distribuição de energia.
- Possíveis causas: accesso indevido à conta de outro utilizado.
- Algumas medidas a implementar/adoptar para eliminar a ameaça: bloqueio da sessão, não afixação de informação confidencial no terminal, confirmação de carregameto multifactor.

7. **Qual dos seguintes componentes (escolher só um) não está relacionado com a tríade CIA?**

Select one:

- A. Integridade
- B. Disponibilidade
- C. Fiabilidade

Correcto! Esta propriedade é mais apropriada para systems de safety. Boa sorte para o resto do exame!

- D. Confidencialidade
- E. Não Repúdio

Confidentiality

It's crucial in today's world for people to protect their sensitive, private information from unauthorized access.

Protecting confidentiality is dependent on being able to define and enforce certain access levels for information. In some cases, doing this involves separating information into various collections that are organized by who needs access to the information and how sensitive that information actually is - i.e. the amount of damage suffered if the confidentiality was breached.

Some of the most common means used to manage confidentiality include access control lists, volume and file encryption, and Unix file permissions.

Integrity

Data integrity is what the "I" in CIA Triad stands for. This is an essential component of the CIA Triad and designed to protect data from deletion or modification from any unauthorized party, and it ensures that when an authorized person makes a change that should not have been made the damage can be reversed.

Availability - although this can be also associated with safety, in the security world it is also applicable, as example:

This is the final component of the CIA Triad and refers to the actual availability of your data. Authentication mechanisms, access channels and systems **all have to work properly** for the information they protect and ensure **it's available when it is needed**.

High availability (HA) systems are the computing resources that have architectures that are specifically designed to improve availability. Based on the specific HA system design, this may target hardware failures, upgrades or power outages to help improve availability, or it may manage several network connections to route around various network outages.