}

**Guide1. APT Project Definition**

**Capstone Subject**

1. **PART I**

|  |
| --- |
| **1. Personal History** |
| Below is a table in which you must complete the requested information. |

|  |  |
| --- | --- |
| Student Name | * **Felipe Torres** * **Manuel Miqueles** * **Nicolas Fernandez** * **Liam Ley** |
| Ruth | * **19.406.513-2** * **20.224.356-8** * **21.193.631-2** * **22.606.024-3** |
| Race | **Computer Engineering** |
| Headquarters | **Duoc UC Puente Alto** |

|  |
| --- |
| **2. APT Project Description** |
| In the description, you must briefly indicate the name of your APT project and the competencies of the graduate profile that you are going to put into practice. If your career defines performance areas, also mention which performance areas the project is linked to. |

|  |  |
| --- | --- |
| Project Name | EncryptU |
| Performance Area(s) | Desktop application programming, web development, computer security. |
| Competences | * Analysis and survey of software requirements. * Programming of secure solutions in multiplatform environments. * Database development and REST APIs. * Quality assurance and project management. |

|  |
| --- |
| **3. APT Project Rationale** |
| Below are different fields that you must fill in with the requested information. This section seeks to describe your project in detail and justify its relevance and relevance. |

|  |  |
| --- | --- |
| Relevance of the APT project | Currently, the Syntec company specializes in the development of technological solutions for institutional and private clients. In the context of the EncryptU project, it seeks to address a growing problem: the lack of accessible and secure tools to protect sensitive information, especially among users with low technological experience.  The selection of the theme responds to the need to create a computer solution that allows the encryption and management of passwords and personal data through a desktop application complemented by a web platform. This approach is highly relevant in the workplace, as it responds to real challenges in cybersecurity, digital accessibility, and data protection, critical areas in modern software development.  The main beneficiaries of the project will be end users such as older adults, students and workers who handle confidential information without advanced technical knowledge. The proposed solution seeks to democratize access to encryption tools, reducing risks of breach and improving personal digital security. |
| APT Project Description | The goal of the EncryptU project is to develop a desktop application with layered architecture that allows you to encrypt, decrypt and manage passwords securely. This solution is complemented by a secondary web application for ticket management and feedback.  The system will feature secure authentication, user and role management, key maintainers, exportable reports, and push notifications. In addition, it will be integrated using REST APIs to ensure interoperability between both platforms. |
| Relevance of the project with the graduate profile | This project is directly related to the profile of graduates of the computer science career, since it requires skills such as:   * Requirements Survey and Analysis * Design and development of computer systems * Implementation of secure technological solutions * Quality Assurance and Functional Testing * Management of IT projects under structured methodologies   These competencies are essential to address the problem raised, allowing a solid technical execution and professional documentation of the system. |
| Relationship to professional interests | Our professional interests are focused on the development of IT solutions that combine security, usability and functional design. The EncryptU project reflects these interests by integrating modern technologies such as AES-256, Django, Oracle SQL and web frameworks, in a realistic development environment.  The experience gained in this project will be key to facing the challenges of the world of work, especially in areas such as cybersecurity, multiplatform development and user experience. |
| Development feasibility of the APT Project | The project will run between August and December 2025, with a total duration of 85 working days, divided into planning, development, testing and closure phases, according to the Gantt schedule.  Available Resources:   1. Personal computer for development and documentation 2. Visual Studio Code (HTML, CSS, JavaScript) 3. Django Framework 4.2.4 with Python 3.10.11, Bootstrap 5 and SweetAlert2 4. Oracle SQL Developer 19c Database   No external factors are identified that hinder development, since in the case the team has previous experience in the selected tools. The detailed planning and modular approach of the system ensure the feasibility of the project within the set deadline. |

1. **PART II**

|  |
| --- |
| **4. Objectives** |
| In this section you must define general and specific objectives of the APT Project. It is important to clarify that the objectives must be stated in a clear, concise way and without giving further explanations, that is, they must be understood by themselves. It is suggested to write them using an infinitive verb, as this makes it necessary to specify specific actions. |

|  |  |
| --- | --- |
| General objective | Develop a cross-platform desktop application with web support that allows non-technical users to protect their information through secure, simple and accessible encryption. |
| Specific objectives | * Implement authentication and role management. * Develop encryption/decryption module with AES/Fernet. * Create user maintainers, passwords, and algorithms. * Generate exportable reports in PDF/XLS. * Integrate REST APIs for interoperability with the secondary web app. * Develop secondary web application for feedback management. * Ensure transport and storage security (TLS + encryption). |

|  |
| --- |
| **5. Methodology** |
| In the next section you must describe the methodology, specific to your discipline, that you will use to solve the APT project described above, including the stages and methods of work. |

|  |
| --- |
| Description of the Methodology |
| To address the identified problem – the difficulty that older adults and non-technical users have to protect their sensitive information with complex encryption tools – the **EncryptU project**  will be developed under the **Waterfall methodology**, which is appropriate given that:   * The problem is clearly defined. * Rigorous control of deliverables within established deadlines is required. * The subject includes structured phases of progress.   **Phases of the methodology**   1. **Planning and requirements gathering**    * Activities: preparation of the Articles of Incorporation, survey of functional and non-functional requirements, definition of use cases.    * Product: Requirements Document (ERS). 2. **Design**    * Activities: creation of mockups, definition of the system architecture (DAS), design of the database model.    * Product: DAS document, model ER. 3. **Development**    * Activities: programming of the desktop application (EncryptU) in Python and the secondary web application with Flask and React, integration with database and REST API.    * Product: Functional desktop application + secondary web application. 4. **Testing and QA**    * Activities: functional testing, security testing (AES/Fernet encryption, authentication), usability testing with representative users.    * Product: Test reports and validation of requirements. 5. **Implementation and Closure**    * Activities: preparation of user manual, final presentation, delivery of documentation and closing minutes.    * Product: User manual, final report, official system delivery.   **Defining roles, tasks, and responsibilities**  The development team will be made up of **5 main roles**, each with specific responsibilities:   * **Project Manager (Felipe Torres)**   + Coordinate and supervise all phases of the project.   + Manage planning, scheduling, and cost control.   + Validate deliverables with the client and stakeholders.   + Lead the final presentation and closing documentation. * **Programmer Analyst (Liam Ley)**   + Develop the desktop application (EncryptU) in Python.   + Implement the encryption/decryption module with AES/Fernet.   + Ensure the integration of the application's internal modules. * **DBA – Database Administrator (Manuel Miqueles)**   + Design and model the database in PostgreSQL/SQLite.   + Implement necessary table scripts and queries.   + Optimize data storage security and performance. * **QA / Tester (Nicolás Fernández)**   + Develop functional and non-functional test plans.   + Run security, usability, and performance tests.   + Document incidents and verify their correctness. * **UI/UX Designer (José Rojas)**   + Create mockups and system interaction flows.   + Ensure the usability and accessibility of the application for non-technical users.   + Design friendly graphical interface with large icons, contrasts and clear texts.   **Focus on the identified problem**  The methodology will allow:   * **To control compliance with the main objective**: to deliver an accessible, simple and secure solution. * **Divide tasks according to specialty**: each member contributes their knowledge (programming, databases, design, QA). * **Avoid ambiguities**: by working with closed phases and clear deliverables, project traceability is ensured. * **Reduce risks**: by carrying out security and usability tests before closing, it is validated that the solution responds to the real need of the target audience (older adults and non-technical users). |

|  |
| --- |
| **6. Evidence** |
| Next, describe what evidence will be evaluated in the progress report and in the final report of your APT project. This evidence must be agreed with your teacher. Evidence will be understood as the products that are developed during the project and whose purpose is to make visible or document how the work has been implemented. |

|  |  |  |  |
| --- | --- | --- | --- |
| Type of evidence | Name of evidence | Description | Justification |
| Advance | Articles of Incorporation | Initial document that defines the objectives, scope, resources, budget, schedule and risks of the project. | Establish the formal basis of the project, delimiting its scope and commitments. |
| Advance | Requirements Form | Spreadsheet with functional and non-functional requirements, prioritization and responsible. | Facilitate traceability and control of requirements throughout the life cycle. |
| Advance | ERS (Software Requirements Specification) document | Document that describes functional and non-functional requirements, use cases and acceptance criteria. | Ensure that the system meets the identified needs of users. |
| Advance | Extended Use Case Document | Detailed use cases with actors, alternative flows, exceptions, and business rules. | Ensure the correct interpretation of interactions between users and the system. |
| Advance | Gantt Chart / Roadmap | Project schedule, in Gantt or roadmap format depending on the methodology. | Plan times, dependencies and those responsible for each phase. |
| Advance | EDT – Costs | Work breakdown structure with estimated cost allocation. | Be clear about the financial resources and man-hours required. |
| Advance | Sprint Progress Document (if applicable Agile) | Record of deliverables and progress achieved in each sprint. | Provide regular visibility into progress in agile methodologies. |
| Advance | Business Process Document (TO-BE) | Model of the business process expected after implementation. | Define how current flows will change with the new system. |
| Advance | Full System Mockups | Visual prototype of screens and application interaction flows. | Validate usability with the client and users before development. |
| Advance | Architecture Document (DAS) | Definition of system architecture: layers, database, REST APIs and technical components. | Ensure scalability, security and good development practices. |
| Advance | Database Model (Scripts) | Table creation scripts. | Ensure consistency and security in information storage. |
| Advance | Data Dictionary | Clear definition of each field, format, unit and data source. | Avoid ambiguity in the interpretation of system information. |
| Advance | RACI Matrix | Table that assigns responsible (R), approvers (A), consulted (C) and informed (I). | Ensure clarity in the assignment of roles and responsibilities. |
| Advance | Risk Matrix | Identification of risks, impact, probability and mitigation plans. | Reduce uncertainty and prepare preventive actions. |
| Advance | Change Control Matrix | Formal registration of change requests with impact on scope, cost and time. | Maintain control over modifications to the project. |
| Advance | Meeting Minutes | Documentation of agreements, commitments and topics discussed in meetings. | Ensure traceability of decisions and commitments assumed. |
| Advance | Test Plan | Document that defines scope, criteria, types of tests and schedule. | Plan and properly organize the validation of the system. |
| Advance | Intermediate Test Reports | Evidence of unit and functional tests carried out in initial phases. | Validate that each module complies before being integrated into the system. |
| Advance | Database Test Matrix | Specific test cases for query validation, integrity, and security. | Ensure the reliability and consistency of the information stored. |
| Advance | Defect Registration Form | Log of errors found, severity, status, and person responsible for correction. | Provide traceability and prioritization to the error correction cycle. |
| Advance | Scope Verification | Comparison document between planned and delivered scope. | Ensure compliance with objectives and detect deviations. |
| Final | EncryptU Desktop Application | Final product in Python with GUI (Tkinter/ttkbootstrap), encryption/decryption modules, authentication and user management. | Deliver concrete solutions for the accessible and secure protection of information. |
| Final | Secondary Web Application | Complementary system in Flask + React for support ticket management and notifications. | Facilitate user feedback and traceability of the use of the system. |
| Final | Training Plan | Document/plan for training end users and system administrators. | Ensure adoption and efficient use of the delivered solution. |
| Final | User Manual | System installation, configuration, and use guide. | Provide clear and accessible documentation for the end user. |
| Final | Project Closure Report | Final document that summarizes achievements, metrics, lessons learned and formalizes closure. | Deliver a global balance of the project and consolidate organizational learning. |

|  |
| --- |
| **7. Work Plan** |
| In the following table define the planning of your APT Project according to what is required. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **APT Project Work Plan** | | | | | | |
| Competence or units of competences | Name of Activities/Tasks | Description Activities/Tasks | Resources | Duration of the activity | Responsible[[1]](#footnote-1) | Remarks |
| Project Management | Articles of Incorporation | Prepare a base document that defines objectives, scope, budget, schedule and risks. | PC, Minutes Templates | 3 | Felipe Torres (Project Manager) | Key document to formalize the start of the project. |
| Project Management | Definition of Requirements (ERS) | Identify and document functional and non-functional requirements. | Interviews, PC, ERS templates | 4 | Felipe + Liam | Requires validation with the customer. |
| Project Management | EDT + RACI Matrix | Work structure breakdown and role assignment. | WBS + Excel Template | 1 | Felipe Torres | It facilitates the management and assignment of tasks. |
| Analysis | SW Architecture Document (DAS) | Definition of layers, APIs, DB and technical components. | PC, modeling software | 3 | Manuel Miqueles (DBA) | It guarantees order and technical quality. |
| Analysis | Use Cases | Model interactions between users and the system. | Use Case Templates | 4 | Felipe + QA (Nicolás) | Input for testing and functional validation. |
| UI/UX Design | Mockups & Prototype | Design interfaces and flows to validate usability. | Figma, PC | 4 | Manuel Miqueles (Designer) | Early validation with the client. |
| Web Development | Web Front-End (Admin) | Construction of an administrative web interface in React. | PC, VSCode, React | 19 | Liam | Requires usability review. |
| Web Development | Web Back-End (Admin) | API and business logic construction in Flask. | PC, Python, Flask | 26 | Liam | Integration with DB and front-end. |
| Desktop Development | Front-End App Desktop | Desktop graphical interface with Tkinter/ttkbootstrap. | Python, VSCode | 19 | Felipe | Oriented to older adults (usability). |
| Desktop Development | Back-End App Desktop | Implementation of internal logic (encryption, user management). | Python, SQLite | 26 | Felipe | Use of OOP and good practices. |
| Safety | AES-256 Implementation and Single Encryption | Development of encryption and decryption modules with Fernet/AES. | Python, cryptography library | 12 | Felipe + Liam | Critical to ensuring data security. |
| Integration | API Implementation | Development and integration of REST services. | Flask, Postman | 12 | Liam | Allows communication between desktop and web. |
| Databases | BD Implementation | Creating tables and queries in PostgreSQL. | SQL Developer, PC | 12 | Manuel Miqueles | Ensure consistency and integrity. |
| UI/UX | Interface Development | Visual adjustments, usability and accessibility. | Figma, Python | 19 | Manuel Miqueles | Adapt to non-technical users. |
| QA / Security | Test Plan | Define functional, non-functional, and safety tests. | PC, documentation | 1 | Nicolás Fernández (QA) | Define metrics and scope of tests. |
| QA / Security | Pentesting and security testing | Vulnerability assessment, encryption, and access. | Kali Linux, PC | 3 | Nicolas Fernandez | Possible adjustments after findings. |
| QA / Security | Usability Testing | Validation with representative users (older adults). | Prototype, real users | 4 | Manuel Miqueles + QA | Measure satisfaction and ease of use. |
| QA | Audit and Preliminary Certification | Comprehensive review of compliance with requirements. | Checklist, PC | 2 | Felipe + QA | Pre-launch requirement. |
| Implementation | Deployment Plan | Phased launch strategy and risk control. | Documentation | 2 | Felipe | Minimizes the impact of failures. |
| Implementation | Phased rollout and Go-Live | Controlled migration and putting into production. | Hosting, servers | 4 | Complete equipment | Critical phase, requires monitoring. |
| Closing | Post-production optimization | Fixes and improvements after Go-Live. | PCs, logs | 7 | Felipe + Liam | System stabilization. |
| Closing | User Manual | Preparation of a step-by-step user guide. | ReportLab, Word | 1 | Jose | Aimed at users without technical knowledge. |
| Closing | Final Report and Formal Closing | Closing document with results, learnings and deliverables. | PC, templates | 1 | Felipe | It marks the official closure of the project. |

|  |
| --- |
| **8. Gantt chart** |
| Look for a Gantt Chart format that suits you and organize the activities planned in the previous point considering the period assigned for the development of your APT Project. You must maintain the temporality of the academic period in the development of the three phases contemplated by the Degree Portfolio Subject. |

|  |  |  |  |
| --- | --- | --- | --- |
| Task Name | Duration | Beginning | The end |
| **Project Initiation** | **85 days** | **Mon 11-08-25** | **Fri 05-12-25** |
| **Planning** | **5 days** | **Mon 11-08-25** | **Fri 15-08-25** |
| Defining Requirements | 3 days | Mon 11-08-25 | Wed 13-08-25 |
| Resource Allocation | 2 days | Mon 11-08-25 | Tue 12-08-25 |
| Requirement Template | 4 days | Mon 11-08-25 | Thu 14-08-25 |
| EDT | 1 day | Mon 11-08-25 | Mon 11-08-25 |
| RACI Matrix | 1 day | Mon 11-08-25 | Mon 11-08-25 |
| Incorporation Act Template | 3 days | Mon 11-08-25 | Wed 13-08-25 |
| Capture of specific requirements | 4 days | Mon 11-08-25 | Thu 14-08-25 |
| SW Architecture Document | 3 days | Mon 11-08-25 | Wed 13-08-25 |
| Use Cases Document | 4 days | Mon 11-08-25 | Thu 14-08-25 |
| Prototype | 4 days | Mon 11-08-25 | Thu 14-08-25 |
| ERS Proposal | 3 days | Mon 11-08-25 | Wed 13-08-25 |
| Risk monitoring | 2 days | Mon 11-08-25 | Tue 12-08-25 |
| Cost Template | 1 day | Fri 15-08-25 | Fri 15-08-25 |
| **Project development** | **65 days** | **Mon 11-08-25** | **Fri 07-11-25** |
| Front-End Web Manager | 15 days | Mon 11-08-25 | Fri 29-08-25 |
| Back-End Web Manager | 20 days | Mon 11-08-25 | Fri 05-09-25 |
| Front-End App Desktop | 15 days | Mon 11-08-25 | Fri 29-08-25 |
| Back-End App Desktop | 20 days | Mon 11-08-25 | Fri 05-09-25 |
| AES-256 Implementation and Single Encryption | 10 days | Mon 08-09-25 | Fri 19-09-25 |
| API Implementation | 10 days | Mon 22-09-25 | Fri 03-10-25 |
| DB Implementation | 10 days | Mon 06-10-25 | Fri 17-10-25 |
| Interface Development | 15 days | Mon 20-10-25 | Fri 07-11-25 |
| **Security Testing** | **8 days** | **Mon 10-11-25** | **Wed 19-11-25** |
| Test Plan | 1 day | Mon 10-11-25 | Mon 10-11-25 |
| Security Testing | 1 day | Tue 11-11-25 | Tue 11-11-25 |
| Pentesting | 2 days | Wed 12-11-25 | Thu 13-11-25 |
| Usability | 2 days | Fri 14-11-25 | Mon 17-11-25 |
| Audit | 1 day | Tue 18-11-25 | Tue 18-11-25 |
| Preliminary Certification | 1 day | Wed 19-11-25 | Wed 19-11-25 |
| **Launch to production** | **6 days** | **Thu 20-11-25** | **Thu 27-11-25** |
| Deployment Plan | 2 days | Thu 20-11-25 | Fri 21-11-25 |
| Phased deployment | 2 days | Mon 24-11-25 | Tue 25-11-25 |
| Go live | 2 days | Wed 26-11-25 | Thu 27-11-25 |
| **Post-Production** | **6 days** | **Fri 28-11-25** | **Fri 05-12-25** |
| Continuous Optimization (Stabilization) | 1 day | Fri 28-11-25 | Fri 28-11-25 |
| Post-implementation audit | 1 day | Mon 01-12-25 | Mon 01-12-25 |
| User Manual | 1 day | Tue 02-12-25 | Tue 02-12-25 |
| Maintenance | 2 days | Wed 03-12-25 | Thu 04-12-25 |
| Formal closure of the project | 1 day | Fri 05-12-25 | Fri 05-12-25 |

Chart, Timeline

AI-generated content may be incorrect.

1. In the event that the APT Project is a group project, in this column they must indicate the name of those responsible for each task or activity. This will later allow differentiating the evaluation by each member. [↑](#footnote-ref-1)