}

1. **PART I**

| **1. Personal Background** |
| --- |

| Student names | Bastián Rodríguez – Daniela Vera – Tamara Fernández – José Miguel Pérez |
| --- | --- |
| National ID (Chile) | 20.397.212-1 21.537.411-4 20.202.332-0 11.656.293-6 |
| Career | Computer Engineering |
| Campus | DUOC UC Puente Alto |

| **2. APT Project Description** |
| --- |

| Project Name | **Sistema de Control de Inventarios para Maestranza Industrial S.A.** |
| --- | --- |
| Performance Areas | * Web application development and implementation. * Database management and modeling. * Information systems security. * IT project management. |
| Competencies | * Design, implement, and maintain information systems. * Manage relational databases. * Develop object-oriented applications using a multi-layered architecture. * Apply security standards and software engineering best practices. * Manage technology projects considering time, resources, and requirements. |

| **3. APT Project Rationale** |
| --- |

| Project Relevance | As a team, we identified that Maestranza Industrial S.A. currently manages its inventory manually, which generates frequent errors, lack of traceability, and operational overcosts. Our project aims to address this situation by developing an automated web-based inventory control system.  This project is relevant because it will optimize critical supply processes, ensure material availability, and improve the organization’s operational efficiency. Furthermore, it allows us to apply our professional competencies in a real-world context aligned with the labor market of our field. |
| --- | --- |
| Project Description | Our goal is to develop a web system with a centralized database that will:   * Register and track inventory movements in real time. * Generate automatic low-stock alerts. * Manage suppliers and internal clients. * Produce dynamic reports exportable in multiple formats. * Implement user roles and profiles for system security.   This way, we seek to deliver a scalable, secure, and efficient solution for the company. |
| Alignment with Graduation Profile | The project is directly related to the graduation profile of our program since it integrates competencies in software development, database management, information security, and project methodologies. All these skills are required to solve the identified problem. |
| Relation to Professional Interests | As a group, we share a common interest in specializing in the development of web systems applied to business management. This project allows us to strengthen those skills, work with current technologies (.NET Core, SQL Server), and apply quality and security standards, thus contributing to our professional growth and preparation for the job market. |
| Project Feasibility | We consider this project feasible because:   * We have the semester timeframe to develop a functional prototype. * The course provides enough hours to progress in each stage. * We have the necessary tools (.NET Core, SQL Server, HTML/CSS, testing environments). * Our team has prior experience in programming and databases. * Potential difficulties (tight deadlines, complex module integration) will be mitigated by prioritizing critical features and applying an agile methodology. |

1. **PART II**

| **4. Objetivos** |
| --- |

| General Objective | To develop and implement a web-based inventory control system for Maestranza Industrial S.A. that optimizes the management of parts and components, improves process traceability, and reduces operational errors. |
| --- | --- |
| Specific Objectives | * Design and model the centralized database of the inventory system. * Implement maintainers for parts, users, suppliers, internal clients, and categories. * Develop dynamic reports exportable to PDF and Excel. * Configure automatic low-stock alerts. * Incorporate authentication and user role control. * Ensure usability and responsive design of the application. |

| **5. Methodology** |
| --- |

| As a team, we will adopt the agile Scrum methodology, organizing ourselves into weekly sprints with partial deliveries and constant reviews. Stages:   1. Requirement gathering and architectural design. 2. Database modeling and definition of use cases. 3. Development of CRUD modules. 4. Implementation of reports and alerts. 5. Integration, testing, and adjustments. 6. Documentation and final delivery. |
| --- |

| **6. Evidence** |
| --- |

| **Type of Evidence** | **Name of Evidence** | **Description** | **Justification** |
| --- | --- | --- | --- |
| Progress | Requirements Matrix and Project Charter | Document defining scope, stakeholders, functional/non-functional requirements. | Ensures early alignment with business needs and guides development. |
| Progress | ER Model + Data Dictionary | Database diagram and glossary of tables/fields/relationships. | Validates data structure before coding and reduces rework. |
| Progress | Multi-layer Architecture + Prioritized Use Cases | System architecture diagram and prioritized CU list. | Demonstrates technical implementation strategy and sprint focus. |
| Progress | CRUD Prototype for Inventory (MVP) | Initial maintainers for parts, categories, and locations with validations. | Provides early functional evidence and enables feedback. |
| Progress | Test Plan + Test Cases | Unit and functional test design with sample dataset. | Guarantees traceability between requirements and verification. |
| Final | Functional Web Application | Full system with authentication/roles, maintainers, alerts, reports, and export features. | Tangible product that fulfills project objectives. |
| Final | REST Service Integration | API consumption/exposure (e.g., suppliers or notifications). | Meets interoperability requirement and enables scalability. |
| Final | Reports with Filters + Export Options | Inventory and consumption history reports exportable to PDF/Excel/CSV. | Adds value for control and auditing. |
| Final | User Manual + Technical Manual | Guides for end-users and system administrators. | Facilitates adoption, support, and maintainability. |
| Final | Final Technical Report + Traceability Matrix | Comprehensive report covering design, implementation, tests, and RF→CU→Test mapping. | Provides formal academic and professional evidence of compliance. |

| **7. Work Plan – APT Project** |
| --- |

| **Competency** | **Task Name** | **Task Description** | **Resources** | **Duration** | **Responsible** | **Observations** |
| --- | --- | --- | --- | --- | --- | --- |
| Project Management (Scrum) | Initial Planning and Charter | Define scope, risks, sprints, and DoD. | Templates | 1 week | Tamara | Basis for monitoring and control. |
| Analysis and Modeling | Requirements and Use Cases | Matrix of functional/non-functional requirements and prioritized use cases. | Templates | 1 week | Tamara + Daniela | Prioritize by value and risk. |
| Database Modeling | ER Model and Dictionary | Logical/physical design, keys, and normalization. | SQL Server | 1 week | José Miguel | Validate with real use cases. |
| Software Architecture | Multi-layer Architecture Definition | Layers (presentation, domain, data), patterns, and security. | .NET Core, Diagrams | 1 week | Bastián | Align with performance requirements. |
| OO Development | CRUD for Parts/Components | Create, read, update, delete with validations and filters. | .NET Core, HTML/CSS | 2 weeks | Bastián + Daniela | Core inventory functionality. |
| OO Development | CRUD for Categories and Tags | Advanced classification of components. | .NET Core | 1 week | Bastián + José Miguel | Supports efficient searches. |
| OO Development | CRUD for Storage Locations | Warehouse structure and traceability. | .NET Core | 1 week | Bastián | Essential for locating items. |
| Information Security | Authentication and Roles | Login, recovery, and role-based authorization. | .NET Identity, SQL Server | 2 weeks | Bastián | Protects critical operations. |
| Interoperability | REST Integration | API consumption/exposure and error handling. | .NET Web API, Postman | 1–2 weeks | Bastián, Daniela, José Miguel | Meets interoperability requirement. |
| Notifications | Stock Alerts and Email | Thresholds, triggers, and notifications. | SMTP/API, Jobs | 1 week | Bastián | Validate with limit cases. |
| Reporting | Reports and Export | Reports with filters and export (PDF, Excel, CSV). | .NET libraries, SQL | 1–2 weeks | Bastián | Must meet ≤ 3s performance. |
| Software Quality | Unit/Functional Testing | Coverage of critical use cases and regression. | Selenium | 1 week | Daniela | Linked to acceptance criteria. |
| UX and Accessibility | Responsive and A11y Adjustments | Validate contrast, keyboard, and responsive design. | HTML/CSS | 1 week | Daniela | Meet accessibility standards. |
| DevOps/Deployment | Deployment and Backup | Publish, automate backups, and recovery. | IIS/Azure, Scripts | 1 week | Bastián + José Miguel | Ensure RTO ≤ 4 h. |
| Documentation | Manuals and Final Report | User/technical manuals and project report with traceability. | Wor | 2 weeks | Team | Final delivery with annexes. |

| **8. Gantt Chart** |
| --- |

| The project schedule is structured over **18 weeks** and divided into three phases:   * **Phase 1 (Planning)**: Weeks 1–4 – Requirement gathering, database design, and architecture. * **Phase 2 (Execution)**: Weeks 5–14 – CRUD development, reports, alerts, authentication, and partial testing. * **Phase 3 (Closure)**: Weeks 15–18 – Integration, final testing, documentation, and final report. |
| --- |

| **Activity** | **Phase 1** | **Phase 2** | **Phase 3** | **Weeks** |
| --- | --- | --- | --- | --- |
| Requirement gathering | ✔ |  |  | 1–2 |
| Database design and architecture | ✔ |  |  | 3–4 |
| CRUD development |  | ✔ |  | 5–7 |
| Role and user management |  | ✔ |  | 8–9 |
| Dynamic reports |  | ✔ |  | 10–11 |
| Automatic alerts |  | ✔ |  | 12–13 |
| Testing and integration |  | ✔ |  | 14 |
| Documentation and final report |  |  | ✔ | 15–18 |