## Penalosa, Mauricio

Developer

Manager: Arjay Gallentes

Evaluated By: Arjay Gallentes

## July Impact Statement

Organization: Axos Business Center Team VII (Arjay Gallentes)

Location: ABC Manila Office

01/01/2025 - 06/30/2025

# Questions

### What do you do? How do you describe your role, duties, and responsibilities? Please be specific.

Manager Evaluation

Response:

Employee Evaluation

Response: As a Developer, I am responsible for designing, building, and maintaining software solutions that align with project and business goals. My duties include writing clean, efficient, and scalable code, participating in code reviews, and collaborating with cross-functional teams to deliver high-quality features. I actively contribute to sprint planning, backlog grooming, and troubleshooting efforts, ensuring timely resolution of issues. I also engage in continuous learning to stay current with technologies and apply best practices in development, testing, and deployment. Additionally, I support knowledge sharing within the team and take initiative in improving development workflows and system performance.

### What processes do you perform? Please refer to the specific process maps or procedures for the tasks you perform, manage, or have impacted. If no process map exists, please describe the task and write "no process map or procedure".

Manager Evaluation

Response:

Employee Evaluation

Response: Backend API Development Process Map

1. Requirement Analysis

- Review user stories or product backlog items.
- Clarify business logic and data flow with stakeholders.
- Identify dependencies (e.g., database schema, external services).
- **Output**: Confirmed API requirements and acceptance criteria.

2. Design

- Define API endpoints, request/response models, and HTTP methods.

- Plan authentication/authorization (e.g., JWT, OAuth).
- Document using Swagger/OpenAPI.
- **Output**: API design spec and interface contract.

**3. Development**

- Set up controller, service, and repository layers.
- Implement business logic using C# and ASP.NET Core.
- Apply SOLID principles and dependency injection.
- **Output**: Functional API endpoints with unit tests.

**4. Testing**

- Write unit and integration tests using xUnit/NUnit.
- Use Postman or Swagger UI for manual testing.
- Validate edge cases and error handling.
- **Output**: Test results and bug fixes.

**5. Code Review**

- Submit pull request with linked work item.
- Peer review for logic, security, and performance.
- Address feedback and refactor if needed.
- **Output**: Approved and merged code.

**6. Deployment**

- Push to staging via CI/CD pipeline (e.g., Azure DevOps).
- Run smoke tests and monitor logs.
- Deploy to production after QA sign-off.
- **Output**: Live API endpoint.

**7. Documentation & Handoff**

- Update API documentation on Confluence.
- Notify frontend or integration teams.
- Log changes in release notes.
- **Output**: Fully documented and supported API.

## What metrics do you use to manage your performance? How do you know you are doing a good job? Please be specific (number of calls per day, number of closed loans per month).

Manager Evaluation ______________________

Employee Evaluation ______________________

Response:

Response: As a Developer, I manage my performance using a combination of task-based metrics and goal tracking aligned with our team's objectives. Specifically:

- **Task Completion and Timeliness**: I track the number of development tasks or user stories completed within a sprint or release cycle, ensuring they meet the defined acceptance criteria and are delivered on time.
- **Code Quality and Review Feedback**: I monitor the number of pull requests approved without major revisions and the quality of peer review feedback to gauge the effectiveness and clarity of my code.
- **Bug Resolution and Defect Rate**: I keep track of bugs reported post-deployment and aim to minimize rework by thoroughly testing and validating my work before submission.
- **Collaboration and Initiative**: I also consider qualitative feedback from peers, especially when I volunteer for additional tasks or contribute to process improvements.

## How do you report progress to your supervisor? (What reports, meetings, etc?)

Manager Evaluation

Response:

Employee Evaluation

Response: I report progress to my supervisor through a combination of structured meetings, documentation, and system updates. I participate in daily stand-up meetings where I share task updates, blockers, and next steps with the team. During sprint retrospectives and planning sessions, I contribute to discussions on what went well, areas for improvement, and action items for the next cycle.

## Which processes / routine tasks have you improved / suggested improvement? What are the specific improvements suggested or made? a. Please refer to specific process maps and procedures for the processes you perform, manage, or have impacted, or if you have created a new process, please describe it. b. If suggested, please outline action steps to implement, include dependencies.

Manager Evaluation

Response:

Employee Evaluation

Response: As a developer, I've focused on identifying and addressing inefficiencies in both technical workflows and cross-functional processes. My contributions have included:

- Enhancing Code Review Efficiency: I introduced a checklist-based peer review process to ensure consistent quality and reduce back-and-forth during pull requests.
- Improving Legacy Code Maintainability: I initiated a refactoring effort for older modules, introducing dependency injection and modular design patterns to improve testability and reduce onboarding time for new developers.

## What other improvements have you suggested (sales, product, group interactions, etc.)? Please be specific.

Manager Evaluation

Response:

Employee Evaluation

Response: I've contributed to key improvements that enhanced both product quality and team efficiency. I supported the integration of Entity Framework 8, repository pattern, services and Fluent API into our .NET tools, improving compatibility and developer experience. I also helped identify and document fixes for issues like thread safety and transaction handling, which improved system stability.

## What improvement opportunities exist to make your role more efficient? What ideas do you have to reduce waste or inefficient work? What are the required action steps?

Manager Evaluation

Response:

Employee Evaluation

Response: Code Optimization & Modern Practices

Use efficient coding patterns such as:

- Avoiding unnecessary object creation and using StringBuilder instead of string concatenation in loops.
- Minimizing LINQ usage in performance-critical paths.
- Leveraging asynchronous programming (async/await) for I/O-bound operations to improve responsiveness

## What improvements have you made or suggested?: How did you positively influence the attitude of your colleagues? Please be specific.

Manager Evaluation

Response:

Employee Evaluation

Response: By improving backend performance and deploying scalable solutions, I’ve likely made it easier for other developers and operations teams to deliver value faster.

### What are your goals? What would you like to accomplish in the next six months? Please make your goals "SMART" - specific, measurable, attainable, relevant, and time-sensitive.

Manager Evaluation

Response:

Employee Evaluation

Response: Enhance API Performance

- **Specific**: Refactor and optimize at least two high-traffic ASP.NET Web APIs to reduce response time.
- **Measurable**: Achieve a minimum 20% improvement in average response time as measured by application monitoring tools.
- **Attainable**: Use profiling tools and async programming techniques.
- **Relevant**: Improves user experience and system efficiency.
- **Time-sensitive**: Complete by December 2025.

Strengthen Unit Testing Coverage

- **Specific**: Increase unit test coverage for core business logic in the UploadFile module.
- **Measurable**: Raise code coverage from current baseline to at least 85%.
- **Attainable**: Allocate 2–3 hours per sprint for test development.
- **Relevant**: Enhances code reliability and supports CI/CD practices.
- **Time-sensitive**: Achieve target by end of Q4 2025.

Improve Documentation and Knowledge Sharing

- **Specific**: Create or update at least three internal documentation pages for recurring development tasks or troubleshooting.
- **Measurable**: Publish docs in the team’s shared knowledge base and present one in a team sync.
- **Attainable**: Use existing experience and recent project learnings.
- **Relevant**: Reduces onboarding time and improves team efficiency.
- **Time-sensitive**: Complete by end of current review cycle.

### What are your goals? What would you like to accomplish in the next 2 years? Goals over the longer term can be more aspirational.

Manager Evaluation

Response:

Employee Evaluation

Response: Over the next two years, my primary goal is to deepen my technical expertise in C# asp.net development, particularly within Azure DevOps and scalable backend systems. I aim to take on more complex development tasks and contribute to architectural decisions that improve system performance and maintainability.

**Long-Term Aspirations:** Looking further ahead, I aspire to grow into a senior developer or solutions architect, where I can influence broader engineering strategy, drive innovation, and help shape the future direction of our platforms and tools.

## Process Maps: Please list the names of all process maps in the iGrafx platform that relate to your specific role. If applicable, please list the names of process maps that relate to your specific role which have yet to be created, so that you get credit for these as well.

Manager Evaluation

Response:

Employee Evaluation

Response: Backend API Development Lifecycle
*Covers:* Requirement analysis, design, implementation, testing, deployment, and documentation.
*Purpose:* Standardize backend development practices and improve handoff between developers and QA.

Security Remediation Workflow
*Covers:* Identification, prioritization, and resolution of penetration test findings.
*Purpose:* Ensure consistent handling of vulnerabilities and compliance with audit requirements.

Legacy System Modernization Process
*Covers:* Assessment, refactoring, and migration of legacy applications (e.g., green screen conversion).
*Purpose:* Guide teams through modernization efforts with minimal disruption.

CI/CD Pipeline for ASP.NET Projects
*Covers:* Build, test, and deployment automation using Azure DevOps.
*Purpose:* Reduce manual errors and accelerate release cycles.

Code Review and Quality Assurance

**Checklist**
*Covers:* Peer review steps, code standards, and readiness criteria.
*Purpose:* Improve code quality and reduce rework.

## Additional Feedback

Additional Feedback: Please feel free to tell us what else you would like us to know about your role in the company, or anything else on your mind.

| Manager Evaluation | | Employee Evaluation | |
|---|---|---|---|
| Response: | | Response: | Thank you for another year and looking forward. |