

## CARRERA DE COMPUTACIÓN

### 1. Datos Informativos

- 1.1. **Módulo:** 1
- 1.2. **Nivel:** 7mo Nivel
- 1.3. **Apellidos y Nombres:** Samira Narváez, Estiven Méndez, Almer Paguay
- 1.4. **Tema:** Normativas 12207/15504 en el proyecto “Recetas en línea”
- 1.5. **Fecha:** 22 de octubre de 2025

### 2. Objetivo

- Finalidad en lograr la disponibilidad del sistema del 99.5% mensual, dándole cero probabilidades de pérdidas de datos de recetas y usuarios.
- Reducir el tiempo para que un nuevo usuario publique su primera receta a menos de 3 minutos (Dándole la medición por test de usabilidad.)
- Garantizar del 95% de las páginas se carguen en menos de 2 segundos, incluso con 10.000 usuarios concurrentes lo cual sería el rendimiento de ello.
- Darle una implementación al proceso de gestión de incidentes que permitan parchear vulnerabilidades críticas identificadas a un plazo de 48 horas como máximo.

### 3. Contenido

Aplicación de las Normativas 15504 y 12207 en el proyecto “Recetas en línea”.

Normativa 15504 y Normativa 12207

Matriz de Evidencias - Proyecto: Plataforma de Recetas en Línea

Evaluación según ISO/IEC 12207 e ISO/IEC 15504

### 1. MATRIZ DE EVIDENCIAS POR PROCESO

#### a) Procesos de Desarrollo



Proceso	Nivel	Aplicación al	Tipo	Ejemplos	Responsable	Estado
según	de	Proyecto de	de	Concretos	able	
ISO	Capa	Recetas	Evidencia			
12207	ciudad					
		Esperado				
<b>Análisis</b>	Nivel	Definir funcionalidad	Documento	- Documento	Product Owner / Analista	
de Requisitos	2 - Gestión	s: registro usuarios, CRUD	Entrevistas	de requisitos funcionales		
	nado	recetas, búsqueda por ingredientes, valoraciones		 - Historias de usuario - Casos de uso documentados - Mockups/wireframes		
<b>Diseño</b>	Nivel	Definir estructura:	Documento	- Diagrama de arquitectura	Arquitecto	
de la Arquitectura	2 - Gestión	Frontend (React/Vue), Backend (Node.js/Django), BD (PostgreSQL/MongoDB), APIs	Técnica	de arquitectura	Software	
	nado			 - Modelo entidad-relación - Diseño de API REST - Flujos de navegación		



**Upec** Impleme

<b>Implementación</b>	Nivel 2 -	Desarrollo de módulos: autenticación, gestión de recetas, buscador, sistema de comentarios	Evidencia Técnica	- Código fuente en repositorio - Commits organizado s - Código revisado (code review) -	Desarrolladores Frontend/ Backend
				Documentación de código	
<b>Integración de componentes</b>	Nivel 2 -	Conectar frontend-backend, integrar pasarela de pagos (si aplica), servicios de imágenes	Evidencia Técnica / Prueba / Pruebas de integración	- Plan de integración  - Pruebas de integración  - Logs de integración  - APIs documentadas	DevOps / Tech Lead
<b>Pruebas y Validación</b>	Nivel 2 -	Pruebas funcionales, usabilidad, rendimiento con muchas recetas,	Documento / Observación	- Plan de pruebas >- Casos de prueba ejecutados - Reporte de	QA Tester / Líder QA

compatibilidad  
móvil bugs<br>-  
Validación  
con  
usuarios  
beta

### b) Procesos de Gestión

Proceso	Nivel	Aplicación	Tipo	Ejemplos	Responsable	Estado
según ISO 12207	de Capacidad	al Proyecto de Espera	de Evidencia Recetas	Concretos	sable	do
<b>Planificación de Proyecto</b>	Nivel 2	Cronograma de sprints, nado	Documento de asignación de tareas, definición de entregables por fase	- Plan de proyecto (Gantt/Kanban) - Actas de reuniones ->- Distribución de roles - Presupuesto estimado	Gerente de Proyecto / Scrum Master	
<b>Gestión de Riesgos</b>	Nivel 2	Identificar riesgos: Gestión de Riesgos	Documento de riesgo: sobrecarga del servidor, problemas de copyright	- Matriz de riesgos - Plan de contingencia - Registro de incidentes ->	Gerente de Proyecto	

**UPEC**

en recetas,  
RGPD/priv  
acidad

<b>Gestión de Configuración</b>	Nivel 2	Control de versiones	Evidencia	- Repositorio Git (GitHub/GitLab)	DevOps / Líder Técnico
	- Gestión nado	del código, gestión de releases, respaldos de base de datos	Técnica	b) - Política de branching >- Versionado semántico >- Changelog documentado	
<b>Medición y Seguimiento</b>	Nivel 3	Métricas de avance, velocity del equipo, tiempo de respuesta de la app, tasa de conversión	Métrica s / dashboard	- KPIs definidos (velocidad, bugs, uptime) - Dashboard de métricas - Reportes semanales >- Análisis de retrospectivas	PMO / Project Manager

### c) Procesos de Soporte

Proceso según ISO 12207	Nivel de Capaci	Aplicación al Proyecto	Tipo de	Ejemplos Concretos	Responsable	Estado
				Calle Antisana y Av. Universitaria Tel. (06) 2980831 - 2984435 info@upec.edu.ec www.upec.edu.ec Tulcán - Ecuador		

dad de Eviden  
Espera Recetas cia  
do

<b>Aseguramiento de Calidad</b>	Nivel 3	Revisiones de código, cido	Documental de auditorías de seguridad , validación de UX/UI	- Política de calidad  Checklist de revisión  - Auditorías internas  >- Estándares de codificación	Respondeble de Calidad
<b>Documentación</b>	Nivel 2	Manual de usuario, nado	Documental de guía de instalació n, documentación técnica (API), guía de contribución	- Manual de usuario  - README completo  >- Documentación API (Swagger) - Wiki del proyecto	Documento lista / Tech Writer
<b>Gestión de Cambios</b>	Nivel 2	Proceso para nuevas funcionalidades	Documento de solicitud de cambios  >- Comité de cambios  >- Registro	- Formulario de solicitud  >- Comité de cambios  >- Registro	Change Manager

usuarios,  
mejoras  
sugeridas  
de  
aprobacione  
s<br>-  
Trazabilidad  
de cambios

<b>Resolución de Problemas</b>	Nivel 2	Atención a bugs reportados	Documental / Técnica	- Sistema de tickets (Jira/Trello)	Soporte Técnico / Desarrolladores
		s, soporte a usuarios, incidencias en producción		 - SLA definido - Base de conocimiento - Logs de resolución	

## 2. CRITERIOS DE EVALUACIÓN ISO/IEC 15504 APLICADOS

Atributos de Proceso por Nivel

### Nivel 1: Realizado (Performed)

El proceso logra su propósito

- Ejemplo: Se desarrollan funcionalidades de recetas, pero sin documentación formal

### Nivel 2: Gestionado (Managed)

- El proceso se planifica y se monitorea
- Los productos de trabajo están identificados y controlados
- Ejemplo: Existe un plan de proyecto con tareas asignadas y control de versiones en Git

### Nivel 3: Establecido (Established)

- Se usa un proceso estándar definido
- El proceso está documentado y se sigue consistentemente
- Ejemplo: Se usa una plantilla estándar para todas las historias de usuario

### Nivel 4: Predecible (Predictable)

- El proceso se ejecuta con límites definidos y se mide

### Nivel 5: Optimizado (Optimizing)

- El proceso se mejora continuamente basándose en datos
- Ejemplo: Se ajusta el proceso de desarrollo según métricas de productividad

## 3. INFORME DE EVALUACIÓN - PROYECTO RECETAS CASERAS

### Datos Generales

- Proyecto: Plataforma Web de Recetas en Línea
- Empresa/Equipo: Samira Narváez, Estiven Méndez, Almer Paguay
- Evaluadores:
- Fecha de evaluación:
- Alcance: Procesos de desarrollo, gestión y soporte según ISO/IEC 12207
- Modelo de evaluación: ISO/IEC 15504 (SPICE)

### Objetivo del Informe

Evaluar el nivel de madurez y capacidad de los procesos de desarrollo de software aplicados al proyecto de plataforma de recetas caseras, según las normas ISO/IEC 12207 (procesos del ciclo de vida) e ISO/IEC 15504 (evaluación de capacidad).

### Resultados de la Evaluación

Proceso Evaluado	Nivel	Fortalezas Detectadas	Oportunidades de Mejora
<b>Análisis de Requisitos</b>	Nivel 1-2	<ul style="list-style-type: none"> <li>• Requisitos funcionales identificados&lt;br&gt;</li> <li>• Participación activa de stakeholders</li> </ul>	<ul style="list-style-type: none"> <li>• Falta documentación formal de requisitos&lt;br&gt;</li> <li>• No hay trazabilidad de cambios</li> </ul>
<b>Diseño Arquitectura</b>	Nivel 2	<ul style="list-style-type: none"> <li>• Arquitectura moderna</li> </ul>	<ul style="list-style-type: none"> <li>• Falta diagrama de despliegue&lt;br&gt;</li> <li>• Sin</li> </ul>

		(microservicios) • Documentación de BD clara	evaluación de escalabilidad
<b>Implementación</b>	Nivel 2	<ul style="list-style-type: none"> <li>Uso de repositorio Git&lt;br&gt;• Code reviews implementados</li> </ul>	<ul style="list-style-type: none"> <li>Estándares de código no documentados&lt;br&gt;• Cobertura de pruebas unitarias baja</li> </ul>
<b>Pruebas y Validación</b>	Nivel 1-2	<ul style="list-style-type: none"> <li>Pruebas funcionales básicas realizadas&lt;br&gt;• Validación con usuarios beta</li> </ul>	<ul style="list-style-type: none"> <li>Sin plan formal de pruebas&lt;br&gt;• Pruebas de carga no realizadas</li> </ul>
<b>Planificación</b>	Nivel 2	<ul style="list-style-type: none"> <li>Cronograma definido&lt;br&gt;• Roles claramente asignados</li> </ul>	<ul style="list-style-type: none"> <li>No hay métricas de seguimiento&lt;br&gt;• Estimaciones sin base histórica</li> </ul>
<b>Gestión de Configuración</b>	Nivel 2	<ul style="list-style-type: none"> <li>Versionado en Git implementado&lt;br&gt;• Estrategia de branching definida</li> </ul>	<ul style="list-style-type: none"> <li>Sin política formal documentada&lt;br&gt;• Falta automatización de despliegues</li> </ul>
<b>Aseguramiento de Calidad</b>	Nivel 1	<ul style="list-style-type: none"> <li>Revisiones de código ad-hoc</li> </ul>	<ul style="list-style-type: none"> <li>Sin política de calidad documentada&lt;br&gt;• No hay auditorías periódicas</li> </ul>
<b>Gestión de Riesgos</b>	Nivel 1	<ul style="list-style-type: none"> <li>Riesgos técnicos identificados informalmente</li> </ul>	<ul style="list-style-type: none"> <li>Sin matriz de riesgos formal&lt;br&gt;• No hay planes de</li> </ul>

### Nivel Promedio de Madurez

Nivel alcanzado: Nivel 2 (Gestionado) - en transición hacia Nivel 3

Los procesos funcionan y están gestionados, pero necesitan mayor estandarización, documentación formal y medición sistemática.

### Conclusiones

1. El equipo ejecuta los procesos fundamentales necesarios para desarrollar la plataforma de recetas
2. Existe gestión básica del proyecto con asignación de roles y seguimiento
3. Falta estandarización: Los procesos varían según quién los ejecute
4. Medición insuficiente: No se recopilan métricas que permitan predecir resultados o mejorar
5. Documentación fragmentada: Información dispersa, sin repositorio central

### Recomendaciones Específicas para el Proyecto:

#### Corto Plazo (1-3 meses)

1. Crear plantilla estándar de historias de usuario con criterios de aceptación claros
2. Implementar un sistema de tickets (Jira/Trello) para gestión de bugs y funcionalidades
3. Definir 5 métricas clave:
  - Velocity del equipo (puntos por sprint)
  - Tiempo de respuesta de la aplicación
  - Tasa de bugs en producción
  - Cobertura de pruebas (objetivo: >70%)
  - Satisfacción de usuarios beta (NPS)
4. Documentar estándares de código (guía de estilo, convenciones de nombres)
5. Establecer Definition of Done para considerar una funcionalidad completa

### Mediano Plazo (3-6 meses)

1. Implementar pipeline CI/CD para automatizar pruebas y despliegues
2. Crear plan formal de pruebas incluyendo casos de prueba de regresión
3. Realizar auditorías de seguridad (validación de inputs, protección de datos personales)
4. Implementar monitoreo en producción (logs, alertas, uptime)
5. Capacitar al equipo en metodologías ágiles y mejores prácticas

### Largo Plazo (6-12 meses)

1. Alcanzar Nivel 3 (Establecido) mediante:
  - Procesos completamente documentados
  - Uso consistente de plantillas y estándares
  - Métricas recopiladas sistemáticamente
2. Implementar mejora continua con retrospectivas y análisis de métricas
3. Crear programa de calidad con auditorías trimestrales

## 4. PLAN DE MEJORA ESPECÍFICO

Acción	Proceso ISO	Responsable	Plazo	Indicador de Cumplimiento (KPI)
<b>Correctiva</b>	<b>12207</b>			
<b>Crear plantilla de requisitos estándar</b>	Análisis de Requisitos	Product Owner	15 días	100% nuevos requisitos con plantilla
<b>Implementar sistema de tickets (Jira)</b>	Gestión de Cambios	Scrum Master	20 días	100% bugs y features en sistema
<b>Definir y medir 5 KPIs del proyecto</b>	Medición	PMO	30 días	Dashboard operativo con datos reales



**POLITÉCNICA  
DEL CARCHI**

EDUCAMOS PARA TRANSFORMAR EL MUNDO

<b>Documentar estándares de código</b>	Implementación	Tech Lead	30 días	Guía publicada + 80% equipo capacitado
<b>Crear plan formal de pruebas</b>	Pruebas y Validación	QA Lead	45 días	Plan aprobado + casos de prueba documentados
<b>Implementar CI/CD básico</b>	Integración	DevOps	60 días	Deploy automático a ambiente de pruebas
<b>Realizar auditoría de seguridad</b>	Aseguramiento de Calidad	Security Officer	60 días	Reporte de vulnerabilidades + plan corrección
<b>Capacitar equipo en SPICE/ISO</b>	Gestión de Calidad	RRHH / Manager	90 días	80% personal certificado o capacitado
<b>Documentar política de calidad</b>	Aseguramiento de Calidad	Responsable Calidad	90 días	Política aprobada y publicada

### Seguimiento y Monitoreo:

- Revisión quincenal de avances en reuniones de equipo
- Evaluación mensual de KPIs con ajustes necesarios
- Re-evaluación formal en 6 meses para medir progreso
- Objetivo a 12 meses: Alcanzar Nivel 3 (Establecido) en procesos críticos

## 5. BENEFICIOS ESPERADOS PARA EL PROYECTO

### Para el Negocio

- Mayor previsibilidad en entregas
- Reducción de bugs en producción

Calle Antisana y Av. Universitaria  
Telf: (06) 2980837 - 2984435  
info@upec.edu.ec  
www.upec.edu.ec  
Tulcán - Ecuador

- Facilidad para escalar el equipo

#### Para el Equipo:

- Claridad en roles y responsabilidades
- Procesos predecibles y documentados
- Menos "apagar incendios"
- Cultura de mejora continua

#### Para la Calidad del Producto:

- Código más mantenible y escalable
- Funcionalidades mejor probadas
- Documentación técnica actualizada
- Seguridad y privacidad garantizadas

### NOTAS IMPORTANTES

Adaptación al contexto: Esta matriz está diseñada para un proyecto de plataforma web. Los niveles de capacidad pueden ajustarse según:

- Tamaño del equipo (startup vs empresa grande)
- Presupuesto disponible
- Complejidad técnica del proyecto
- Madurez actual de la organización

Enfoque ágil: Las normas ISO son compatibles con metodologías ágiles (Scrum/Kanban). La clave es adaptar la documentación y procesos para que agreguen valor sin burocratizar.

Mejora incremental: No es necesario alcanzar Nivel 5 en todos los procesos. Prioriza los procesos críticos para tu proyecto y avanza gradualmente.

## 4. Conclusiones

- Las normas ISO 12207 E ISO 15504 se diría que son un marco de trabajo probado para proporcionar la estructura necesaria para que “Recetas en Línea” evolucioné en una idea a un producto escalable y de mayor calidad.

a aplicación de estos estándares podrá garantizar que la calidad se

"agregue "en el producto desde las primeras etapas, evitando costosos cambios futuros y construyendo una base técnica sólida.

- Siendo el caso la mejora continua estará en la complementariedad, donde La 12207 define "**que hacer**" en los procesos y la 15504 mide "**que tan bien lo hacemos**", dándole un ciclo de mejora continua basado en datos.
- Acoger la disciplina desde el inicio diferencia a "Recetas en Línea" de los competidores permitiéndose crecer de forma ordenada y un alta Recompensa del usuario al final.

## 5. Referencia Bibliografía

### Bibliografía

Ingertec. (s.f.). *Ingertec*. Recuperado el 22 de octubre de 2025, de ISO 15504 SPICE: <https://ingertec.com/listado-normas-iso/iso-15504/>

Ingertec. (s.f.). *Ingertec*. Obtenido de ISO 12207. Ciclo de Vida del Software: <https://ingertec.com/iso-iec-12207/>

Ruiz, D. J. (s.f.). *Scribd*. Obtenido de ISO/IEC 15504 SPICE: <https://es.scribd.com/document/539832991/Iso-iec-15504-Spice>

Wikipedia. (s.f.). *Wikipedia*. Obtenido de ISO/IEC 12207: [https://es.wikipedia.org/wiki/ISO/IEC\\_12207](https://es.wikipedia.org/wiki/ISO/IEC_12207)

## 6. Referencia linkografía

- [¿Iso 15504?](#)
- [¿Iso 12207?](#)
- [¿15504 Spice?](#)
- [¿ISO/IEC 12207?](#)