



**Uptc**<sup>®</sup>  
Universidad Pedagógica y  
Tecnológica de Colombia



# Pautas de Proyecto

**Segundo Semestre 2022**

Ingeniería del Software II

Este material de autoestudio fue creado en el año 2022 para la Escuela de Ingeniería y Computación de la Seccional de Sogamoso y ha sido autorizada su publicación por el (los) autor (es), en el Banco de Objetos Institucional de la Universidad Pedagógica y Tecnológica de Colombia.

**EDMUNDO ARTURO JUNCO ORDUZ**

[edmundo.junco@uptc.edu.co](mailto:edmundo.junco@uptc.edu.co)

Docente - Escuela de Ingeniería de Sistemas y Computación

Universidad Pedagógica y Tecnológica de Colombia

Grupo de Investigación Galash

Docente UPTC – Sogamoso

Tel 3102852189

# PROYECTO

## Ingeniería del Software II

Señor@s estudiantes

Reciban un cordial saludo, por medio de esta información se presenta las Pautas establecidas para el proyecto final de la Asignatura.

- Documento IW

Glosario

Generalidades.

1. Introducción

1.1 Descripción del problema.

1.2 Formulación.

1.3 Justificación.

1.4 Estado de Arte. O antecedentes

1.5 Objetivos

1.5.1 General.

1.5.2 Específicos.

2. Marco Teórico.

2.1 Marco referencial

2.2 Marco Legal (si se requiere)

2.3 Marco Conceptual.

3. Metodologías.

3.1 Tipo De Metodología (Investigación – si se requiere)

3.2 Metodología del Software (Roles).

3.2 Modelo - método

3.3 Sistema Actual.

3.4 Sistema propuesto.

4. Requisitos. (solicitados por la Metodología)

4.1 Historias de Usuario (solicitados por la Metodología)

5. Diseño de software (solicitados por la Metodología)

5.1 Modelo entidad relación Físico, Lógico.

5.2 Diccionario de datos.

6. Prototipo del Software

6.1 Prototipos de Administrador

6.2 Prototipos de Usuario

7 Uml2 (Los solicitados por la Metodología).

8. Arquitectura de Software (Físico, Lógico)

8.1 **Idoneidad Funcional**

Complejidad Funcional

Exactitud Funcional

Pertinencia Funcional

### **Eficiencia de ejecución**

Tiempo a comportamiento - pedido / respuesta – tiempo esperado / tiempo máximo tolerado

Uso de recursos – Que se necesita aprovechar los recursos (**hardware App** - memoria, cpu, disco) (**Horas pico**)

Capacidad – tolerancia o recursos que el sistema necesita para su buen funcionamiento. (Hardware y Software). **Limite de tolerancia / limite tolerancia esperado**

## **Compatibilidad**

Interoperabilidad – Microservicio – la fecha  
Coexistencia – BD

## **Usabilidad**

Reconocimiento de idoneidad  
Curva de aprendizaje  
Operabilidad  
Protección de errores  
Estética de interfaz = UI Y UX  
Accesibilidad =

## 8.2 Compatibilidad

## **Confiabilidad**

Madurez  
Disponibilidad  
Tolerancia de fallos  
Capacidad de recuperación

## **Seguridad**

## **Mantenibilidad**

## **Portabilidad del producto**

### 8.3 Atributos de calidad (Tensión de atributos)

### 8.4 Patrones de arquitectura

### 8.5 EventSourcing

### 8.6 Patrones: Microservicios

### 8.7 Hexagonal - Puertos y adaptadores

### 8.8 Diseño orientado al dominio

### 8.9 Administración Cliente Servidor

### 8.10 Consumo Web

## 9 CONSTRUCCIÓN

### 9.1 Lenguaje de Construcción

### 9.2 Fundamentos de la Construcción del Software

#### 9.2.1 Biblioteca de Construcción

### 9.3 Repositorios (Fechas)

### 9.4 Gestión de la configuración

### 9.5 Reutilización de Código

## 9 PRUEBAS

### 10. PRUEBAS DE SOFTWARE (Verificación / Validación)

#### 10.1 Pruebas de Base de Datos - (Webapp)

#### 10.2 Prueba de Usabilidad -(Webapp) formato de usabilidad

#### 10.3 Prueba de Estadístico y dinámicas.

#### 10.4 Auditoria y reporte de las pruebas (UDMINER - ACS)

#### 10.5 Pruebas Modernas

- 10.6 Repositorio Administración de la Configuración del Software (ACS)
- 10.7 Capas del proceso ACS
- 10.8 Administración de cambios (4 Clases).
- 10.9 Varios – Pruebas anexos.
- 11. METRICAS
  - 11.1 Métricas y Mediaciones
  - 11.2 Métricas basadas en funciones
  - 11.3 Métricas orientadas a clases M-CK (mpc- pah-ndh)
  - 11.4 Métricas de diseño para WebApps
  - 11.5 Métricas de Código Fuente
  - 11.6 Métricas de Pruebas
  - 11.7 Métricas Mantenimiento.
- 12. Mantenimiento de Software (Información posible de Segunda Versión)
  - 12.1 Proceso
  - 12.2 Técnicas
- 13. Calidad de Software (Si es necesario)
- 14. BYOD and MDM (Web) (Si es necesario)
- 15. IBM- Bonita (Si es necesario)
- 16. Estándares.
- 17. Conclusiones
- 18. Bibliografía
- 19. Infografía

#### Actividades Complementarias

- Manual de programador
- Manual de usuario
- Aplicativo terminado 100% y Documento.
- Ponencia del proyecto (Exposición)
- Video – proyecto de aplicativo
- Artículo del proyecto (Tiempo)
- **Aplicación debe estar montada (Dominio y Hosting o APK) Gratuito**

Cordialmente,

**M.Sc. Edmundo Arturo Junco Orduz**

**Docente - Escuela de Ingeniería de Sistemas y Computación**

**Universidad Pedagógica y Tecnológica de Colombia**

**Docente UPTC – Sogamoso**

