

Proyecto Integrador Global

Objetivo

El alumnado optimizará el proceso, desarrollo y liberación de componentes de software mediante la integración de metodologías, modelos, herramientas y servicios para la implementación de aplicaciones WEB empresariales seguras.

Descripción

Desarrollar una aplicación web que permita a las instituciones educativas de nivel media superior, dar seguimiento al cumplimiento de la planeación didáctica de los docentes, así como al control de avances en cada parcial, y la evidencia de capacitación docente durante un periodo académico.

Funcionalidades Principales

1. Gestión de Planeación Didáctica

- Los profesores podrán registrar y subir su **planeación didáctica** por materia y parcial.
- La aplicación permitirá a los coordinadores revisar, aprobar o solicitar ajustes en la planeación.
- Historial de planeaciones anteriores de ciclos anteriores para consulta.

2. Control de Avances por Parcial

- Cada profesor registrará el **avance logrado en cada parcial** con respecto a su planeación inicial.
- El sistema permitirá indicar si el avance fue **cumplido, parcial o no cumplido**.
- Generación de reportes visuales (gráficas de cumplimiento por profesor, materia y parcial).

3. Evidencias de Curso y Capacitación

- Los profesores podrán adjuntar **evidencias de cursos o talleres** tomados durante el periodo (archivos PDF, imágenes, constancias).
- Cada evidencia tendrá campos de: nombre del curso, institución emisora, fecha, horas acreditadas.
- Validación y aprobación por parte de la coordinación académica.

4. Reportes y Seguimiento Institucional

- Reportes globales para la coordinación y dirección:
 - Porcentaje de cumplimiento de planeaciones.
 - Cumplimiento de avances por parcial.
 - Cursos de actualización docente acreditados por periodo.
- Exportación de reportes en PDF o Excel.

Entregables del Proyecto

I. Definición del proceso de desarrollo Web

- ✓ **Seleccionar y justificar la metodología ágil.** - Se mencionan al menos cinco puntos clave de justificación de la metodología ágil de acuerdo a requerimientos de proyecto.
- ✓ **Definir esquema de pruebas.** – Definir en la fase de inicio las pruebas a tomar en cuenta, las pruebas son **pruebas unitarias** y **pruebas funcionales**.
- ✓ **Presentar planeación del proceso** (roles, iteraciones, entregables).- Presentar el plan con Github projects: Actividades, responsable(s), período de ejecución de la actividad , estado y ligado al ISSUE en Github.
- ✓ **Justificar la arquitectura de software elegida.** - La arquitectura propuesta para el desarrollo web se debe basar en una **arquitectura multicapa** (presentación, lógica de negocio, datos, integración y pruebas).
- ✓ **Elaborar un diagrama de arquitectura.** - Indicar las capas de la arquitectura, el flujo de datos, los patrones de diseño y entorno de ejecución
- ✓ **Seleccionar y justificar patrones de diseño** (estructurales, creacionales, de comportamiento y emergentes) Justificar mediante propuesta de la aplicación, al menos seis patrones de diseño que aseguran escalabilidad, mantenibilidad, seguridad y facilidad de integración.
- ✓ **Frameworks Web utilizados.** - Al menos justificar dos frameworks tanto para el FrontEnd como del BackEnd.
- ✓ **Instalar y configurar el entorno de desarrollo** (ej. Node.js, React, Spring Boot, etc.).

II. Control de Versiones

- ✓ **Seleccionar y configurar plataforma de versionamiento Git y GitHub.**
- ✓ **Definir estrategias de:** Nomenclatura de ramas, acceso de usuarios, políticas de merge, políticas de Pull Request.
- ✓ **Push a través de Pull Request.**

Global

- ✓ Aplicar al menos tres procesos de automatización mediante Github WorkFlows (pruebas, modificación, aceptación de pull Request, etc)
- ✓ Demostrar control de versiones desde: línea de comandos, IDE, Github.

III. Integración de componentes

- ✓ Implementar mecanismos de seguridad con HTTPS, JWT y OAuth2.
- ✓ Integrar al menos tres API de terceros (ejemplo: Google Maps, PayPal, OpenWeather).
- ✓ Implementar al menos cinco Web Services de su autoría (REST o SOAP).

IV. Pruebas y liberación

- ✓ Desplegar la aplicación en contenedores Docker tanto para el FrontEnd como el BackEnd.
- ✓ Configurar infraestructura en la nube a través de cuentas gratuitas o de estudiante con empresas como AWS, Azure, Google Cloud, etc (servidor de aplicaciones, servidor de BD, balanceador, dominio, certificados SSL).
- ✓ Realizar pruebas de software con herramientas seleccionadas (ej. Postman, Jest, JUnit).
- ✓ Utilizar herramienta de escaneo de código con una herramienta al gusto que cumpla los principios de OWASP con respecto a la seguridad de código. Cumplir con los diez principios de OWASP

V. Documentación y Entrega

- ✓ Portada (nombre del proyecto, alumno, carrera, materia, docente, fecha).
- ✓ Índice estructurado.
- ✓ Introducción (descripción general del sistema, problemática que resuelve, justificación de la metodología ágil seleccionada).
- ✓ Desarrollo
- ✓ Metodología de desarrollo ágil, arquitectura del software, frameworks, esquema de control de versiones, planeación, diagrama de arquitectura
- ✓ Integración de componentes de software (APIs, Web Services, seguridad, autenticación, etc.).
- ✓ Infraestructura y despliegue (servidores, contenedores, nube, certificados, balanceo de cargas).
- ✓ Pruebas de software.
- ✓ Conclusiones y aprendizajes.
- ✓ Bibliografía bajo norma APA 7.

Global

La entrega del proyecto debe incluir:

- ✓ **Código fuente completo**
- ✓ **Evidencias de funcionamiento** (video corto o demo).
- ✓ **Contenedor o despliegue en la nube** (Dockerfile, docker-compose o configuración en servicios cloud).

Global

Rúbrica de Evaluación Global mediante Cumplimiento Total de Tareas

Criterio	Excelente (100%)	Muy Bueno (80%)	Bueno (70%)	Suficiente (60%)	Deficiente (40% o menos)	Ponderación
I. Definición del proceso y arquitectura SABER	Selección y justificación de manera sólida metodología ágil, definición de esquema pruebas completo, planeación completa, arquitectura de la aplicación y su diagrama; diagrama completo y patrones bien aplicados. Justificación completa de frameworks	Justificación de la metodología clara, pero con pequeños vacíos; patrones parcialmente correctos. Arquitectura y diagrama completo. planeación completa. Justificación completa de frameworks.	Justificación de metodología aceptable, pero con omisiones importantes; diagrama básico. Planeación completa. Justificación de frameworks parcialmente correcto	Presenta metodología y arquitectura sin justificación clara. Planeación completa. Justificación de frameworks incompleto.	No presenta metodología, pruebas ni arquitectura. Justificación de arquitectura y frameworks deficiente.	20%
II. Control de versiones SABER	Configuración completa de plataforma, ramas, PR, accesos; evidencias en CLI, IDE y herramienta gráfica. Workflows completos. Issues enlazados a Github projects	Configuración adecuada con evidencia parcial (faltan una o dos herramientas). Workflows completos. Issues enlazados a Github projects.	Flujo de Git básico, con evidencia mínima. Solo dos workflows. Sin comentarios claros. Issues no enlazados a GitHub	Uso limitado de Git, sin políticas claras. Issues no enlazados a GitHub	No presenta control de versiones. Issues no enlazados a Github Projects	20%
III. Integración y seguridad SABER- HACER	Integra API´s completa y Web Service propio con seguridad robusta; justificación técnica clara.	Integra solo dos API y Web Service con seguridad parcial.	Integra solo una de las dos (API o Web Service) con seguridad básica.	Integración mínima, sin seguridad adecuada.	No integra ni API ni seguridad.	20%
IV. Pruebas y liberación SABER- HACER	Contenedores configurados correctamente; despliegue en nube con BD, balanceo, dominio y certificados; pruebas manuales +	Contenedores y nube configurados parcialmente; pruebas manuales completas.	Despliegue limitado en nube; solo pruebas manuales. Faltan al menos cuatro principios de OWASP	Despliegue incompleto; pruebas poco claras. Faltan al menos seis	No realiza pruebas ni despliegue.	20%

Global

Criterio	Excelente (100%)	Muy Bueno (80%)	Bueno (70%)	Suficiente (60%)	Deficiente (40% o menos)	Ponderación
	automáticas. Cumplimiento completo de OWASP	Faltan al menos dos principios de OWASP		principios de OWASP		
V. Documentación y entrega SABER- HACER	Documento y presentación muy claros, evidencias completas, vídeos o demos correctos y completos.	Documento y presentación adecuados con algunos detalles. vídeos o demos correctos y completos.	Documento y presentación básicos, con poca profundidad. Vídeos medianamente entendibles	Documento poco organizado, evidencias mínimas. Vídeos medianamente entendibles	No entrega documentación ni presentación. Vídeos incompletos	10%
VI. Asistencia a asesorías SABER- HACER	Se agendan y asiste al menos cinco asesorías vía SIGA	Se tiene cuatro asistencias, participación adecuada.	Se tiene tres asistencias, participación adecuada.	Se tiene dos asistencias, participación adecuada.	Se tiene una o menos asistencias, participación adecuada.	10%