

Evaluación B. Proyecto

Objetivo general: aprender las actividades fundamentales de la ingeniería de software, mediante el desarrollo de un proyecto real con objetivos de calidad y tecnologías en el estado del arte.

Objetivos entrega 1

- Establecer un repositorio para el trabajo en equipo y procedimientos de versionado
- Aplicar el proceso de ingeniería de requerimientos para un problema real
- Generar una especificación detallada y bocetos de interfaz de usuario

Objetivos entrega 2

- Construir la aplicación usando las tecnologías definidas
- Desarrollar la interfaz de usuario siguiendo principios de usabilidad y accesibilidad
- Aplicar prácticas de calidad de código y test unitario
- Realizar test de sistema
- Reportar issues y realizar evaluación global de calidad

Las consultas del proyecto deben plantearse únicamente por el foro de Aulas, de forma de que todos los participantes del curso puedan ver las respuestas. No se responderán consultas dos días antes de la fecha de entrega.

Entregas

Para el desarrollo del proyecto se creará un repositorio GitHub a cada equipo. La documentación se realizará en formato Markdown.

Checkpoints. Durante el desarrollo del proyecto se realizan entregas y revisiones de avance. Las entregas para cada checkpoint deben ser subidas en GitHub en la semana indicada del curso. Se dará feedback en clase sobre el avance de los proyectos.

Guía de avance	Semana
Elicitación y análisis	5
Checkpoint obligatorio 1 Elicitación y especificación	7
Bocetos IU y validación de requerimientos	9
Entrega Informe académico 1	
Checkpoint obligatorio 2 IU + clases del dominio	12
Codificación y test unitarios	13

3 Code freeze	15
Testing funcional (proyecto asignado)	16
Entrega Informe académico 2	

Informe académico. En cada entrega se realiza un informe académico que debe dar evidencia de las actividades realizadas y explicar la aplicación de buenas prácticas de ingeniería de software.

Trabajo individual. En el informe se deben detallar las actividades realizadas por cada integrante del equipo: fecha, actividad, responsable, esfuerzo en horas. Cada integrante del equipo debe realizar actividades en todas las áreas indicadas en la rúbrica de cada entrega.

Defensa oral. Se seleccionarán equipos para realizar una defensa del proyecto. Se debe realizar una presentación oral de 10 minutos que presente las principales técnicas aplicadas y aprendizajes. Se realizarán preguntas individuales de profundización y autoría.

Tecnologías

- Repositorio GitHub
- Documentación formato Markdown
- Runtime JavaScript: Node.JS
- Interfaz de usuario: web / mobile (responsive)
- Librería de IU: Bootstrap
- Estándar de accesibilidad: WCAG
- IDE: Visual Studio Code
- Estándar de codificación HTML/JS: Prettier
- Análisis estático de código: ESLint
- Framework de test unitario: Jest
- Reporte de issues en GitHub

Se dará apoyo en tecnologías a través de Teams: Ayudantía FIS.

Problema objetivo

El problema objetivo del proyecto será definido por los docentes y comunicado por Aulas. Se seleccionará un problema real con posibilidad de acceder a usuarios finales para realizar investigación de campo y validación de resultados.

La investigación del problema y delimitación del alcance es responsabilidad de los estudiantes. Se deben especificar en forma detallada los requerimientos que permitan alcanzar los objetivos identificados.

Para la etapa de implementación se deben priorizar las funciones que agregan mayor valor a los usuarios. Alcance mínimo: desarrollar 3 user stories (cada US deben aportar valor al usuario en forma independiente). La implementación no debe tener login ni contraseña de ningún tipo.

Evaluación entrega 1

Área	Objetivos	Puntos
Repositorio Git	Creación y uso de repositorios locales y remotos Comandos Git ejecutados desde terminal	1
Versionado	Aplicar buenas prácticas de versionado Uso de ramas separadas de 'main' Resumen de commits y evolución del proyecto	2
Elicitación	Evidencia de actividades de investigación Referencias a fuentes de información Caracterización de usuarios: User Personas Modelo conceptual del problema	2
Especificación	Definición de requerimientos funcionales y no funcionales User Stories / Use Cases detallados Bocetos de IU	2
Checkpoint obligatorio 1	Elicitación y especificación de requerimientos Se evaluará de acuerdo al grado de avance	4
Validación y verificación	Verificar la especificación Validar la solución con personas no involucradas en el proyecto	2
Reflexión	Detalle del trabajo individual Técnicas aplicadas y aprendizajes	2

Evaluación entrega 2

Área	Objetivos	Puntos
Construcción	Implementación de funciones principales (sin la necesidad de persistencia de datos) Uso de librerías externas y scripts	4
Interfaz de usuario	Interfaz de usuario web / mobile (responsive) Buenas prácticas de implementación de frontend Aplicar un sistema de diseño y principios de usabilidad Estándar de accesibilidad WCAG	4
Codificación	IDE Visual Studio Code: configuración común del equipo Estándares de codificación HTML, JavaScript Buenas prácticas de OOP: separación de lógica e interfaz Análisis estático de código: mostrar reducción de problemas	4
Checkpoint obligatorio 2	Codificación de la IU y clases del dominio Se evaluará de acuerdo al grado de avance	4
Test unitario	Test unitarios en Jest 100% cobertura en clases de dominio	4
Code freeze	En la semana previa a la entrega se debe congelar el desarrollo. A partir de este punto solo se realizan actividades de test de sistema, reporte de issues y generación de informes. Si se modifica el código pasada la fecha afectará el puntaje.	
Test de sistema	Realizar test de sistema (se asigna otro proyecto) Definición de ambientes de testing Generar casos de prueba con técnicas de caja negra Detallar sesiones de prueba exploratoria	3
Reporte de issues	Reportar issues (bugs, improvements, missing features) en GitHub Aplicar buenas prácticas de reporte de issues Definir labels para tipos de issue y niveles de severidad	3
Informe de calidad del sistema	Sumarizar número de issues reportados por tipo Realizar una evaluación global de la calidad	2
Reflexión	Detalle del trabajo individual Técnicas aplicadas y aprendizajes	2

Requisitos de entrega

Identificación. El nombre del repositorio debe contener el apellido de todos los estudiantes del equipo, en orden alfabético, sin caracteres especiales. Por ejemplo: gonzalez-rodriguez. Adicionalmente, cada informe debe contener la identificación de los estudiantes en el encabezado.

Repositorio Git. Todos los elementos del proyecto y su historial de versiones estarán contenidos en un repositorio GitHub (documentación, código, ejecutables y otros recursos). Cada estudiante debe usar un usuario propio para representar su trabajo específico en el proyecto. Para la corrección solo se consideran los elementos de la rama 'main'. No se pueden realizar modificaciones en el repositorio luego de la fecha y hora de cierre definida por la Universidad. La gestión y respaldos del repositorio son responsabilidad de los estudiantes.

Informe académico. El informe académico de cada entrega debe realizarse en un archivo del repositorio usando el formato Markdown. El informe académico incluye las secciones de la rúbrica de cada entrega. En el informe se pueden incluir links a otros archivos del repositorio.

IMPORTANTE. Sistema de gestión ORT. Para entregar todos los obligatorios los estudiantes deben estar inscriptos y formar un equipo en el sistema de gestión de la Universidad. Antes de la fecha de cierre se debe subir un archivo al sistema con el informe académico. Si no se cumple con estos requisitos administrativos no se puede corregir la entrega.

Segunda instancia

Los estudiantes que no lleguen a 70 puntos podrán entregar una segunda instancia del obligatorio 2. Para esta instancia se aplica la misma letra y rúbrica que en la instancia anterior. La entrega en segunda instancia es individual. En caso de aprobar la segunda instancia la nota final del curso será de 70 puntos. Se debe crear un nuevo repositorio para versionar el desarrollo de la segunda instancia. Como punto de partida se puede usar entrega anterior del proyecto. Se debe actualizar la documentación con los cambios realizados y entregar un nuevo informe académico con el detalle del trabajo individual. La defensa de la segunda instancia se coordina con el docente del curso.