

# Fundamentos de Ingeniería de Software

## Proyecto 2022-03 *NFTs* (Lectura 21-mar-22)

**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

## 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. Incluir en el informe una reflexión sobre las técnicas de ingeniería de software y los aprendizajes del proyecto.

Trabajo individual: detallar las actividades realizadas por cada integrante del equipo, en el informe académico y mediante commits en el repositorio. Cada integrante debe realizar actividades en todas las áreas indicadas en esta letra.

Defensa oral: realizar una presentación oral de 10 minutos que presente las principales técnicas aplicadas y aprendizajes.

## Tecnologías

- Repositorio GitHub
- Documentación formato Markdown
- Runtime JavaScript: Node.JS
- Interfaz de usuario: web / mobile (responsive)
- Sistema de diseño: Material Design
- IDE Visual Studio Code
- Test unitario Jest

## Desafío del proyecto

*NFTs* es una aplicación que permite a los usuarios comprar, vender y descubrir artículos digitales.

Se trata de un desafío abierto, se deben explorar funciones que sean de valor para los usuarios. La investigación del problema y delimitación del alcance es responsabilidad de los estudiantes. Para la implementación, se deben priorizar las funciones que agregan mayor valor a los usuarios.

Las consultas del obligatorio (sobre alcance de la letra o sobre tecnologías) deben plantearse en MS Teams: Ayudantía FIS (canal: Consultas Proyecto).

### Evaluación entrega 1 (02-may-22) (25 puntos)

Área	Objetivos	Puntos
Repositorio Git	Creación y uso de repositorios locales y remotos Comandos Git ejecutados desde terminal y desde el IDE	2
Versionado	Aplicar buenas prácticas de versionado Uso de ramas separadas de 'main' Resumen de commits y evolución del proyecto	3
Elicitación	Evidencia de actividades de investigación Referencias a fuentes de información Caracterización de usuarios: User Personas Modelo conceptual del problema	5
Especificación	Definición de requerimientos funcionales y no funcionales User Stories / Use Cases detallados Bocetos de IU	10
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	3

### Evaluación entrega 2 (27-jun-2021) (35 puntos)

Área	Objetivos	Puntos
------	-----------	--------

Construcción	Implementación de funciones principales (sin la necesidad de persistencia de datos) Configuración de plataforma tecnológica para desarrollo y producción Documentación del uso de librerías externas (package.json)	5
Interfaz de usuario	Interfaz de usuario web / mobile (responsive) Página única con navegación entre secciones Implementación: Material Design Web Components Aplicar un sistema de diseño y principios de usabilidad Cumplimiento de estándar de accesibilidad WCAG Seguir especificación de estilo	5
Codificación	IDE Visual Studio Code: configuración común del equipo Estándares de codificación Google (HTML, CSS, JavaScript) Buenas prácticas de OOP: separación de lógica e interfaz Análisis estático de código: mostrar reducción de problemas	5
Test unitario	Test unitarios en Jest 100% cobertura en clases de dominio	5
Code freeze	En la semana previa a la entrega se debe congelar el desarrollo (22-jun-2021). A partir de este punto solo se realizan actividades de test de sistema, reporte de issues y generación del informe académico.	
Test de sistema	Realizar test de sistema sobre la versión congelada Ambiente de testing independiente del desarrollo Generar casos de prueba aplicando técnica partición equivalente Detallar sesiones de prueba exploratoria	5
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 Dejar issues abiertos para correcciones o mejoras futuras Sumarizar número de issues reportados por tipo Realizar una evaluación global de la calidad	5
Reflexión	Técnicas aplicadas y aprendizajes del equipo. Evidencia del trabajo en equipo.	5

# Especificación de estilo

Nombre de la app: *NFTs*

Colores primarios y secundarios seleccionados de: <https://material.io/resources/color>

Font: Roboto

Icons: filled

## Procedimiento de entrega

### Repositorio Git

La creación del espacio de trabajo colaborativo se coordinará con el docente del curso.

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 de cierre. La gestión y respaldos del repositorio son responsabilidad de los estudiantes.

### Informe académico

El informe académico de cada entrega debe realizarse en un archivo README.md utilizando el estándar Markdown (<https://github.github.com/gfm/>). El informe académico debe incluir las mismas secciones de la letra. Desde el archivo README.md se pueden incluir links a otros archivos del repositorio.

### 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, el archivo README.md debe contener información identificatoria en el encabezado.

### Repositorio GitHub

<https://classroom.github.com/a/ZoKB9i1c>

### Template de estructura del Proyecto

<https://github.com/ORT-FIS-202108/Proyecto>

### Sistema de gestión ORT

**IMPORTANTE.** 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. Si no se cumple con estos requisitos administrativos no se puede corregir la entrega.

La versión final del archivo README.md debe integrarse a la rama 'main' y subirse en formato pdf al sistema de gestión de la Universidad.