### Fundamentos de Ingeniería de Software

## Proyecto 2022S2 *Mundial*

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.

### Presentación en clase

Cada equipo deberá realizar una presentación en clase sobre una práctica o tecnología de ingeniería de software aplicada en el proyecto. Se coordinará el tema y la fecha de presentación con el docente.

Mostrar ejemplos de aplicación en el proyecto y recomendaciones útiles para los otros equipos. Duración aproximada de la presentación: 30 minutos.

### 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

*Mundial* es una aplicación que permite presentar el fixture y registrar una predicción para los partidos del mundial.

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 (3-oct-2022)

|  |  |  |
| --- | --- | --- |
| **Área** | **Objetivos** | **Puntos** |
| Repositorio Git | Creación y uso de repositorios locales y remotos  Comandos Git ejecutados desde terminal y desde el IDE | 5 |
| Versionado | Aplicar buenas prácticas de versionado  Uso de ramas separadas de 'main'  Resumen de commits y evolución del proyecto | 5 |
| 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 | 5 |
| Validación y verificación | Verificar la especificación  Validar la solución con personas no involucradas en el proyecto | 3 |
| Reflexión | Detalle del trabajo individual  Técnicas aplicadas y aprendizajes | 2 |

### Evaluación entrega 2 (code freeze: 21-nov-2022 / test otro proyecto: 28-nov-2022)

|  |  |  |
| --- | --- | --- |
| Área | Objetivos | Puntos |
| Construcción | Implementación de funciones principales (sin la necesidad de persistencia de datos)  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.  A partir de este punto solo se realizan actividades de test de sistema, reporte de issues y generación de informes. |  |
| Test de sistema | Realizar test de sistema (se asigna otro proyecto)  Ambiente de testing independiente del desarrollo  Generar casos de prueba aplicando técnica partición equivalente  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 | 2 |
| Informe de calidad del sistema | Sumarizar número de issues reportados por tipo  Realizar una evaluación global de la calidad | 3 |
| Reflexión | Detalle del trabajo individual  Técnicas aplicadas y aprendizajes | 2 |
| Presentación en clase | Práctica o tecnología de ingeniería de software aplicada en el proyecto  Mostrar ejemplos de aplicación y recomendaciones útiles  Duración aproximada de media hora | 5 |

## Especificación de estilo

<https://material.io/resources/color>

Nombre de la app: *Mundial*

Colors primary: Green 900, secondary: Light Blue 800

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.

**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.