

INGENIERÍA DE SOFTWARE 2

Ingeniería en Informática

Facultad Politécnica

Universidad Nacional de Asunción

Guía para el Proyecto de Software

Cuerpo Docente

Profesor de Cátedra	Prof. Lic. Guillermo González, Mst. PMP
Auxiliar de Enseñanza	

Autores: Prof. Lic. Guillermo González, Mst. PMP
ggonzalez@pol.una.py

2024

Versión 1

Introducción

Contenido

Objetivo del proyecto de software

Organización de los equipos de proyecto

Hitos del proyecto

Check Point: Presentación de la asignatura

Check point: Formación de equipos

Hito 1: Revisión y ajuste de ERS (Especificación de Requerimientos de software)

Entregable

Hito 2: Diseño UML (Unified Modeling Language)

Entregables

Hito 3: Sprint 1 del desarrollo de software

Check Point: Planificación del Sprint 2

Entregable

Hito 4: Sprint 2 del desarrollo de software

Hito 5: Sprint 3 del desarrollo de software

Hito 6: Sprint 4 del desarrollo de software

Hito 7: Sprint 5 del desarrollo de software

Hito 8: Sprint 6 del desarrollo de software

Metodología de trabajo

Evaluación

Introducción

El objetivo de esta guía de procedimientos es definir el proceso del desarrollo del proyecto de software a realizar por los alumnos de Ingeniería del Software 2 como parte fundamental de las prácticas de dicha asignatura, organizando y coordinando las actividades a realizar por los alumnos y profesores.

Este proyecto de software consiste en el desarrollo de un sistema software, utilizando el ciclo de vida adaptado de la Metodología SCRUM, debido a la naturaleza de los equipos de trabajo, del tamaño del proyecto y el tiempo disponible en el curso académico.

Con esta guía, se pretende conseguir, además:

- *A nivel didáctico:* Servir como información básica que guíe a los alumnos en el proceso de desarrollo de sus tareas en el marco de un proceso de software definido.
- *A nivel organizativo:* Mejorar la coordinación, organización, planificación y ejecución de las actividades entre los miembros del equipo de trabajo involucrados así como también los profesores de la asignatura.

Contenido

Esta guía de procedimientos incluye lo siguiente:

- *Objetivo del proyecto de software:* Descripción del objetivo de la realización del proyecto de software planteado.
- *Organización de los equipos de proyecto:* Descripción general de los Equipos de proyecto.
- *Hitos del proyecto:* Descripción de cada hito del proyecto y de los procedimientos correspondientes para el logro de los mismos. Calendario de los hitos. Entregables.
- *Metodología de trabajo:* Descripción de la manera de interactuar entre profesores y alumnos para el desarrollo del proyecto.
- *Evaluación:* Descripción y explicación del método de cálculo para las calificaciones de los diferentes equipos.

Objetivo del proyecto de software

Para el proyecto de software a desarrollar se tendrá en cuenta sólo un tema, que el profesor divulgará en la plataforma y en clases presenciales. Normalmente, este será el mismo tema del proyecto desarrollado en la asignatura Ingeniería de Software I dictada en el semestre inmediato anterior a este curso.

Este proyecto de software está orientado no solo a realizar el proceso de software hasta la fase de implementación de software sino también que el proyecto de software en sí ayude al alumno a conocer más sobre la ingeniería de software.

En este caso, el tema del proyecto de software es un *Sistema de Administración de Contenido Web (CMS)* que pretende que los alumnos “*aprendan haciendo*” los conceptos y la

importancia de un área fundamental en la ingeniería de software en la actualidad: Trabajo en equipo con metodologías ágiles y tiempos definidos. Las especificaciones originales de Ingeniería de Software I están en este enlace: https://drive.google.com/file/d/1TFok_pshdVgJJmYiR5IqtYpYdnnv6P0em/view?usp=drive_link

En ese sentido, además de realizar un proyecto software con enfoque más académico (en el sentido de lo explicado en el párrafo anterior) también se busca que el alumno aplique en el proceso de software herramientas actuales que dan soporte al proceso de software, potenciando la Gestión de la Ingeniería de Software.

Otro objetivo no menor es que los alumnos conformen un *equipo de proyecto* considerando la gran diferencia existente entre lo que es un *grupo*. Lo que se pretende en este sentido es que los alumnos entiendan y experimenten otros aspectos no relacionados a la ingeniería de software pero que estarán presente siempre en cualquier proceso de software de su vida profesional: la interacción entre el equipo humano del proyecto y las habilidades necesarias en esta área de gestión de proyectos para lograr el objetivo del proyecto.

Organización de los equipos de proyecto

Los diferentes integrantes de los equipos se denominará como se indica en la siguiente tabla:

Siglas	Denominación	Formado por:
PO	Product Owner	El profesor de la asignatura
DT	Development Team	Alumnos de la asignatura que realizan el proyecto
SM	Scrum Master	Un alumno de cada DT que realiza el papel de Scrum Master

Se debe considerar que las siglas utilizadas en la tabla anterior serán utilizadas a lo largo del desarrollo del proyecto.

Para aprovechar mejor la experiencia de este proyecto, los DT estarán conformados por hasta 4 (cuatro) alumnos de la asignatura, de manera a formar un equipo compacto pero completo.

Hitos del proyecto

Para el logro de los objetivos del proyecto se establecieron los siguientes hitos, según el calendario académico de la asignatura:

Hito	Descripción
cp	Presentación de la asignatura
cp	Formación de equipos de trabajo
cp	Revisión y ajuste de ERS
cp	Elección de IDE, servidor de SCM

1	Entrega de ERS y Epics
cp	Revisión de Diseño
2	Entrega de Diseño
3	Sprint 1 del desarrollo del software
cp	Planificación Sprint 2
4	Sprint 2 del desarrollo del software
cp	Planificación Sprint 3
5	Sprint 3 del desarrollo del software
cp	Planificación Sprint 4
6	Sprint 4 del desarrollo del software
cp	Planificación Sprint 5
7	Sprint 5 del desarrollo del software
cp	Planificación Sprint 6
8	Sprint 5 del desarrollo del software (3er parcial) Sprint 6 del desarrollo del software (1er final)
9	Sprint 6 del desarrollo del software

Para verificar fechas de entrega referenciar a siguiente link:
<https://docs.google.com/spreadsheets/d/1c8ZDFXM5L9CENIIUphzOFyYMt8XihEPSG522wK2Y1N0/edit?usp=sharing>

A continuación se exponen los hitos del proyecto y la descripción del alcance de los mismos.

Check Point: Presentación de la asignatura

En el primer día de clases se realizará la presentación de la asignatura y la modalidad de trabajo, de manera a predisponer a los alumnos a la metodología de evaluación que será procesual, considerando puntos como la completitud y puntualidad de los entregables del proyecto de software.

Check point: Formación de equipos

Se procederá a formalizar la conformación de los equipos comunicando al PO por el medio acordado, de manera a poder identificar y especificar formalmente los miembros del equipo en la plataforma.

Los equipos no podrán cambiar su composición hasta el final de la asignatura, por lo que un desmembramiento o división no será considerado como justificación para el re-acomodo de los miembros en otros equipos. Esta condición es mandatoria.

Hito 1: Revisión y ajuste de ERS (Especificación de Requerimientos de software)

En base a la especificación inicial del proyecto, y tras las consultas y cambios correspondientes al respecto, el DT debe redactar un documento de gestión de cambios del ERS, donde se expongan los requerimientos de software del sistema propuesto contemplando los cambios solicitados por el PO. Este debe contener los siguiente:

Citar requerimientos nuevos. Justificar

Citar requerimientos cambiados . Justificar

Citar requerimientos eliminados. Justificar

Listado de EPICs con Historias de Usuarios en el JIRA

Este documento debe mantenerse actualizado a medida que avanza el proyecto con los cambios solicitados por el PO, de manera a tener referencia futura sobre los cambios incorporados al proyecto. Se definirá un lugar en la plataforma para alojar y mantener actualizado el documento.

Los criterios a ser evaluados son los siguientes:

BAS: Línea Base: entrega requerimientos anteriores.

NCR: Nota de Cambio de requerimientos: nuevos, eliminados, modificados (justificación)

ACT: Nueva versión 01 de ERS actualizado y entregado

COH: Coherencia de requerimientos especificados

APL: Aplicación de los cambios funcionales solicitados en los requerimientos.

FOR: Formato del documento: numeración de requerimientos, títulos, subtítulos, ordenamiento, agrupación de requerimientos.

QA: Aseguramiento de la calidad del entregable (Factor de ponderación sobre el promedio de los criterios anteriores)

Entregable

A continuación las instrucciones para el entregable del hito por equipo:

1. Se definirá un lugar en la plataforma para alojar y mantener actualizado el documento.
2. El nombre del archivo debe estar en el siguiente formato:

EQUIPO_08_A_NCR_01

Donde:

- 08 es el número de equipo. Deben reemplazar éste número por el número de su equipo.
- A es el código de ordenamiento
- NCR es el tipo de documento (Nota de Cambios en ERS)

- 01 es la versión del documento
- Todas las letras en mayúscula

Para documentar la Línea Base de los requerimientos, el equipo debe entregar la especificación de requerimientos utilizados como línea base. El nombre del archivo debe estar en el siguiente formato:

EQUIPO_08_A_ERS_00

Donde:

- 08 es el número de equipo. Deben reemplazar éste número por el número de su equipo.
- A es el código de ordenamiento
- ERS es el tipo de documento (Especificación de Requerimientos de Software)
- 00 es la versión del documento (Línea Base)
- Todas las letras en mayúscula

La primera versión del documento de especificación de requerimientos de software también debe entregarse. El nombre del archivo debe estar en el siguiente formato:

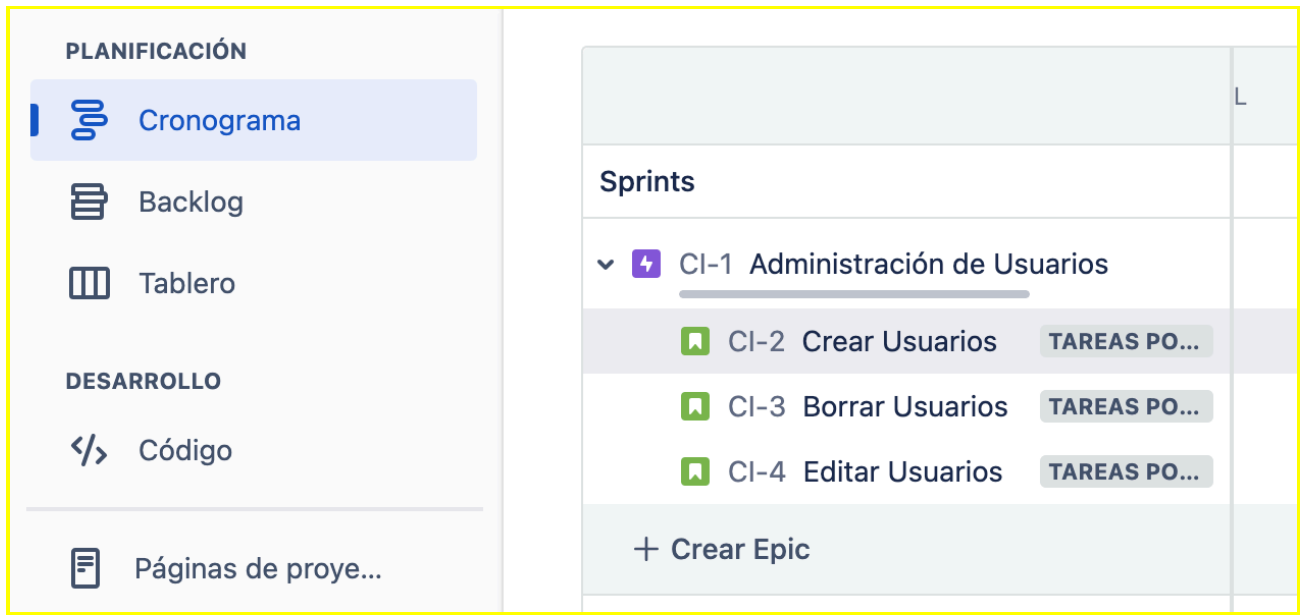
EQUIPO_08_A_ERS_01

Donde:

- 08 es el número de equipo. Deben reemplazar éste número por el número de su equipo.
- A es el código de ordenamiento
- ERS es el tipo de documento (Especificación de Requerimientos de Software)
- 01 es la versión del documento (Primera versión)
- Todas las letras en mayúscula

La primera versión del listado de EPICs con Historias de Usuarios cargadas en [JIRA](#) teniendo en cuenta:

1. Listado de Epics
2. Historias de Usuarios
3. Roles del sistema
4. Criterios de aceptación de cada historia de usuario



En esta etapa aún no se va a considerar la planificación por sprint.

Hito 2: Diseño UML (Unified Modeling Language)

El Diseño UML se realizará en la herramienta de modelado a definir en clase. Es obligatorio cumplir con las indicaciones de la versión a utilizar del software para evitar incompatibilidad al momento de la evaluación del trabajo.

En este hito es importante tener bien definido y claro la plataforma de desarrollo a utilizar ya que el diseño dependerá de este detalle no menor. Los DT deben investigar, antes de llegar a este hito, sobre los requerimientos no funcionales y arquitectura de software que deben modelar en los diferentes diagramas solicitados.

En este hito se debe entregar el proyecto realizado en la herramienta, con los diagramas actualizados según los cambios incorporados por el PO y agregando los diagramas siguientes:

1. Clases + Paquetes (Django) (pueden ser 2 diagramas separados)
2. Componentes (Django + entornos) (1 diagrama por entorno desarrollo y producción)
3. Despliegue (Django + entornos)(1 diagrama por entorno desarrollo y producción)

Los criterios a ser evaluados son los siguientes:

CLA: Diagrama de clases y paquetes adecuados al framework de desarrollo de software

COM: Diagrama de componentes adecuados al framework de desarrollo de software y a los ambientes de desarrollo y producción

DES: Diagrama de despliegue adecuado al framework de desarrollo de software y a los ambientes de desarrollo y producción.

FRA: Aplicación de conceptos del framework de desarrollo de software al diseño del proyecto
AMB: Ambientes de desarrollo y producción contemplados en todo momento en el diseño de software
QA: Aseguramiento de la calidad del entregable (Factor de ponderación sobre el promedio de los criterios anteriores)

Entregables

A continuación las instrucciones para el entregable del hito por equipo:

Diseño

1. Se debe tener 1 (un) sólo documento digital de DIS (diseño de software) del proyecto.
2. Este documento debe ser el proyecto de diseño realizado en la herramienta especificada.
3. Cada diagrama será exportado a formato PDF para ser agregado al .zip

El nombre del archivo debe estar en el siguiente formato:

EQUIPO_08_B_DIS_01.zip

Donde:

- 08 es el número de equipo. Deben reemplazar éste número por el número de su equipo.
- B es el código de ordenamiento
- DIS es el tipo de documento (Diseño de Software)
- 01 es la versión del documento (Versión 1)
- Todas las letras en mayúscula

Para documentar la Línea Base del diseño, el equipo debe entregar el diseño utilizado como línea base. El nombre del archivo debe estar en el siguiente formato:

EQUIPO_08_B_DIS_00.zip

Donde:

- 08 es el número de equipo. Deben reemplazar éste número por el número de su equipo.
- B es el código de ordenamiento
- DIS es el tipo de documento (Diseño de Software)
- 00 es la versión del documento
- Todas las letras en mayúscula

Hito 3: Sprint 1 del desarrollo de software

En esta entrega el DT deberá culminar el Sprint 1 del proceso de desarrollo de software considerando los siguientes puntos a ser evaluados:

IDE : Montaje y utilización del IDE (Integrated Development Environment) con las funcionalidades automáticas configuradas (Ej. Git, Pyunit o pruebas unitarias, documentación de código, otros)

SCC : Sistema de control de cambios (Subversion, GIT, Mercurial, otros) montado y en funcionamiento. Utilización del **Tag** para el primer Sprint, marcando en el trunk del repositorio principal.

PUN : Configuración, funcionamiento y utilización de pruebas unitarias del código fuente del framework

PDO : Configuración, funcionamiento y utilización de documentación de código fuente generado automáticamente

AMB : Ambientes de producción y desarrollo, montados y funcionando.

PLA: Entregar la planificación de historias de usuario para el siguiente sprint en la plataforma JIRA

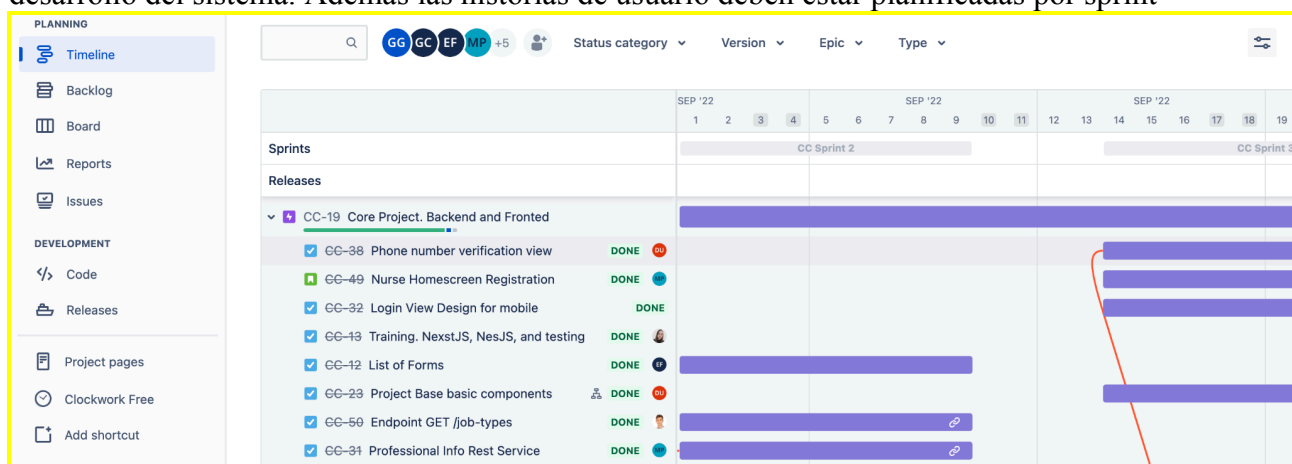
QA: Calidad del proceso y el software desarrollado para el Sprint.

Check Point: Planificación del Sprint 2

Entre uno de los ejercicios que los DT deben realizar está el de la planificación de los entregables de las iteraciones de desarrollo de software a realizar (Sprints). Este trabajo comprende el análisis de la meta a lograr (especificada por el PO) y los trabajos que debe realizar el DT para llegar a lograr la meta. El proceso de planificar es uno de los más enriquecedores para los ingenieros de software, ya que nos ayuda a calcular de alguna manera con los mejores datos disponibles al momento cuánto trabajo debemos hacer y en que tiempo para lograr la meta.

Entregable

1. Plataforma [JIRA](#) con la lista de historias de usuario, epics y sprints necesarios para el desarrollo del sistema. Además las historias de usuario deben estar planificadas por sprint



Hito 4: Sprint 2 del desarrollo de software

En esta entrega el DT deberá culminar el Sprint 2 del proceso de desarrollo de software considerando los siguientes puntos a ser evaluados:

SCC : Utilización de sistema de control de cambios con TAGs

PUD : Utilización de pruebas unitarias y documentación de código fuente

ALC : Alcance a lograr del proyecto de software:

- Administración de Categorías
- Administración de Contenido (solo creación)
- Administración de Usuarios
- Administración de Roles

PLA : Evaluación de la planificación realizada para el sprint actual y la planificación para el siguiente sprint en la plataforma JIRA

QA : Calidad del proceso y el software desarrollado para el Sprint

Además, se debe entregar la planificación de desarrollo de historia de usuarios para el siguiente Sprint.

Hito 5: Sprint 3 del desarrollo de software

En esta entrega el DT deberá culminar el Sprint 3 del proceso de desarrollo de software considerando los siguientes puntos a ser evaluados:

SCC : Utilización de sistema de control de cambios con TAGs

PUD : Utilización de pruebas unitarias y documentación de código fuente

ALC : Alcance a lograr del proyecto de software

- Administración de flujos/estados de contenido
- Administración de Contenido (completa)

PLA : Evaluación de la planificación realizada para el sprint actual y la planificación para el siguiente sprint en la plataforma JIRA

QA : Calidad del proceso y el software desarrollado para el Sprint

Además, se debe entregar la planificación de desarrollo de casos de uso para el siguiente Sprint.

Hito 6: Sprint 4 del desarrollo de software

En esta entrega el DT deberá culminar el Sprint 4 del proceso de desarrollo de software considerando los siguientes puntos a ser evaluados:

SCC : Utilización de sistema de control de cambios con TAGs

PUD : Utilización de pruebas unitarias y documentación de código fuente

ALC : Alcance a lograr del proyecto de software

- Kanban (completo)
- Almacenamiento en la nube de imágenes y otros archivos
- Notificaciones por email
- Búsqueda y filtros en la home page
- Historial de cambios de publicaciones

PLA : Evaluación de la planificación realizada para el sprint actual y la planificación para el siguiente sprint en la plataforma JIRA

QA : Calidad del proceso y el software desarrollado para el Sprint

Además, se debe entregar la planificación de desarrollo de casos de uso para el siguiente Sprint.

Hito 7: Sprint 5 del desarrollo de software

En esta entrega el DT deberá culminar el Sprint 5 del proceso de desarrollo de software considerando los siguientes puntos a ser evaluados:

SCC : Utilización de sistema de control de cambios con TAGs

PUD : Utilización de pruebas unitarias y documentación de código fuente

ALC : Alcance a lograr del proyecto de software

- Comentarios
- Reacciones
- Compartir
- Calificación

PLA : Evaluación de la planificación realizada para el sprint actual y la planificación para el siguiente sprint en la plataforma JIRA

SCR: Script de despliegue automático del proyecto en los ambientes de desarrollo y producción (código fuente, base de datos y población de base de datos)

QA : Calidad del proceso y el software desarrollado para el Sprint

Además, se debe entregar la planificación de desarrollo de casos de uso para el siguiente Sprint.

Hito 8: Sprint 6 del desarrollo de software

En esta entrega el DT deberá culminar el Sprint 6 del proceso de desarrollo de software considerando los siguientes puntos a ser evaluados:

SCC : Utilización de sistema de control de cambios con TAGs

PUD : Utilización de pruebas unitarias y documentación de código fuente

ALC : Alcance logrado del proyecto de software

- Favoritos Categoría
- Contenido destacado
- Generar reportes estadísticos de las publicaciones (interacciones, visualizaciones, visitas)
- Funcionamiento correcto de script de despliegue automático para entorno de producción

INF: Informe Final del Proyecto

El informe ejecutivo a entregar en la Iteración 6 debe tener en resumen las características del proyecto y el proceso de software utilizado y debe estar redactado en lenguaje formal. Deberá contar con la siguiente estructura base:

- Carátula
- Introducción
- El equipo de trabajo, descripción de perfiles
- Descripción del proceso utilizado en el proyecto
- Oportunidades de mejora en el desempeño del equipo
- Lecciones aprendidas del proceso
- Experiencias importantes a mencionar
- Conclusiones

Debe entregarse un informe por equipo el día del examen final en un archivo PDF, subido al Drive del equipo y con un máximo de 6 (seis) páginas.

QA : Funcionamiento correcto del software desarrollado para el Sprint

- Revisión general del flujo crítico de creación y administración de contenido
- Manejo de Roles
- Home y buscador
- Revisión de Sprint en JIRA con todas las historias de usuario creadas y redactadas.
- Asignación de Tareas

Metodología de trabajo

El proyecto de la asignatura será realizado en la modalidad presencial, con la tutoría del profesor asignado.

Desde el primer entregable, el DT está siendo evaluado con el objetivo de tener una calificación al final del curso, de manera que se califica el proceso realizado más que tener un sistema funcional sin haber cumplido los hitos establecidos.

Cada entregable será evaluado en las fechas de entrega y esa calificación formará parte del puntaje final de la asignatura obtenido por los miembros del DT.

Respecto a las consultas que tengan los DT respecto al desarrollo del proyecto, en la plataforma utilizaremos mucho la comunicación asíncrona (Foros, mails y demás). También serán realizadas las aclaraciones en clase, en las revisiones o en los días de consultas.

Es posible que esta planificación requiera de algunos ajustes a ser realizados en el momento de su ejecución, por lo que los alumnos deben estar atentos a las directivas que el profesor realiza en las clases o en la plataforma de manera a cumplir de la manera especificada finalmente.

Evaluación

El proyecto de la asignatura Ingeniería de Software 2 será evaluado en los diferentes aspectos establecidos para los entregables descritos a continuación y a su puntaje.

Examen	Evaluación	Entregable	Puntaje
1er Parcial	Teoría	Examen escrito (TEO)	30
	Proyecto	Sprints del proyecto software (SPS)	70
	TOTAL		100
2do Parcial	Teoría	Examen escrito (TEO)	30
	Proyecto	Sprints del proyecto software (SPS)	70
	TOTAL		100
Final	Teoría	Examen escrito (TEO)	30
	Proyecto	Sprints del proyecto software ponderadas (SPSP) $SPSP = \text{Promedio}(\text{Sprints 1 al 5}) \times 0,4 + \text{Sprint 6} \times 0,6$	70
	TOTAL		100

Recursos

Instalar Django

<https://docs.djangoproject.com/en/4.2/intro/install/>

Django SSO con Google

<https://pypi.org/project/django-google-sso/>

Tutorial Django

<https://docs.djangoproject.com/en/4.2/intro/tutorial01/>