



# PROYECTO DE PASAJE DE GRADO BT DE INFORMÁTICA Edición 2026

Sistema de Gestión de Residuos Urbanos  
SiGeRU

- 1. Introducción al documento**
- 2. Conformación de Grupos**
- 3. Entregas de los avances del Proyecto**
  - 3.1. Consideraciones**
  - 3.2. Fechas
- 4. Defensa del Proyecto Final**
- 5. Generalidades del Proyecto**
- 6. Detalle de Implementación**
- 7. Requisitos generales**
- 8. Herramientas y ambiente**
- 9. Requerimientos por Unidad Curricular**
  - 9.1. Ingeniería de Software**
  - 9.2. Proyecto UTU LAB**
  - 9.3. Administración de Sistemas Operativos**
  - 9.4. Programación Full Stack**
  - 9.5. Emprendedurismo y Gestión**
- 10. Anexos**

## 1. Introducción al documento

---

En esta instancia se desarrolla brevemente el contenido de dicho documento referente al desarrollo del proyecto de manera integradora, identificando en tres hitos parciales, como entregas de carácter de seguimiento y control del mismo.

Este proyecto aplica a los terceros años del EMT de Informática, de las orientaciones **Desarrollo Web y Soporte y Desarrollo**.

## 2. Conformación de Grupos

---

- El proyecto será realizado idealmente por grupos de tres a cuatro alumnos, nunca cinco, y en acuerdo con el docente de Proyecto y bajo su autorización, quedando a criterio del cuerpo docente y de la coordinación de informática habilitar alguna otra excepción como grupos de 2 alumnos.
- Cada grupo de proyecto deberá elegir entre sus integrantes a un Coordinador de Proyecto que tendrá la representación oficial del grupo a los efectos de cualquier notificación, trámite o indicaciones técnicas. Se solicitará también Subcoordinador, a modo de segundo interlocutor, previendo circunstancias coyunturales que pudieran presentarse, y que inhabilitarían la participación del Coordinador.
- El grupo de proyecto deberá seleccionar un Nombre de Empresa (nombre fantasía) a los efectos de identificar con una denominación única al grupo, siendo esta acción obligatoria.
- La presentación de la documentación del proyecto se realizará siguiendo las pautas que oportunamente se suministrarán.
- Cada grupo confirmará su integración oficial, mediante un formulario en el sitio web de la escuela ([https://www.esi.edu.uy/conformacion\\_grupos](https://www.esi.edu.uy/conformacion_grupos)), indicando el grupo de clase, nombre de fantasía de la empresa, así como la nómina de los integrantes con C.I., nombre, teléfono, e-mail de cada integrante donde además se deberá identificar al Coordinador del Proyecto y al Subcoordinador, y agregar el reglamento interno del grupo”
- Esta gestión tendrá como plazo máximo el estipulado más abajo.
- Cada grupo deberá entregar al docente de proyecto el reglamento interno de funcionamiento, donde deberán explicitar mínimamente los siguientes puntos:

- Motivos que se considerarán válidos para la expulsión del grupo.
- Qué sucedería con el nombre de la empresa, en caso de expulsión o disolución del grupo.
- Una vez que los grupos han sido establecidos en tiempo y forma, **cualquier cambio en su integración, deberá ser gestionado por escrito ante el cuerpo docente y la coordinación de informática, y no será válido hasta recibir una confirmación afirmativa.**
- Los grupos deberán respetar la letra del proyecto en todos sus términos y en especial lo que respecta a la conformación de grupos, esto supone que aquellos grupos que no fueron autorizados a cambiar su conformación por parte de la coordinación deberán continuar trabajando como fueron creados.
- Para que un grupo pueda solicitar la expulsión de un integrante de su equipo, se deberá hacer una carta firmada por todos los integrantes, y enviarla al docente de proyecto y a **esi.buceocoordinacion@gmail.com** explicando los motivos que llevaron a dicha solicitud. Además se deberán anexar copia del reglamento interno y la documentación que avale tal pedido con el fin de analizarlo.
- Ante la expulsión de un alumno de un grupo, el grupo deberá continuar trabajando con los integrantes que queden en él.
- Ante la posibilidad de que algún grupo quede con menos de tres integrantes por motivos distintos a la expulsión, la coordinación y el profesor de proyecto serán los que decidan que se hará con los integrantes de dicho grupo.
- La fusión de grupos no está permitida
- Cualquier solicitud de cambio de grupo deberá plantearse entre la Primera y la Segunda Entrega, y debe estar acompañada de documentación que avale el pedido.
- Para el caso de que un alumno decida no continuar realizando el proyecto este año deberá:
  - Llenar y enviar desde su correo a coordinación y al docente de proyecto la carta correspondiente donde deje explícito que abandona la realización del proyecto
  - Aceptar que para poder hacerlo deberá dejar por lo menos una asignatura del tronco tecnológico para el año siguiente porque no se puede hacer el proyecto si no cursa por lo menos una de ellas.
- **En caso de separación de un equipo, expulsión o movimientos de un integrante de un equipo a otro, la documentación generada hasta ese momento, será propiedad**

**de cada uno de los participantes. Esta norma tendrá prioridad sobre cualquier reglamento interno.**

*“El proyecto debe ser realizado únicamente por los integrantes del grupo establecido. Se entiende que compartir total o parcialmente cualquier actividad del proyecto atenta contra la integridad del estudiante y de su formación, por lo tanto, constituye una falta grave. Específicamente no es posible compartir por ninguna vía entre integrantes de grupos distintos, las tareas de codificación, digitación, compilación, depuración y documentación de los programas u objetos (o entregas) del proyecto. Además de que no se pueden compartir actividades del proyecto, no se pueden compartir los productos de las mismas. Cada grupo es responsable del trabajo de su proyecto y de que el mismo sea individual, independientemente de las causas que pudiesen originar la no individualidad. A modo de ejemplo, y sin ser exhaustivos: utilización de código realizado en cursos anteriores (por otros estudiantes) u otros cursos, perder el código, olvidarse del código en lugares accesibles a otros estudiantes, prestar el código o dejar que el mismo sea copiado por otros estudiantes, dejar la terminal con el usuario abierto al retirarse, enviarse código por mail, utilizar código suministrado por terceros, etc. En caso de constatar dicha falta pasará a ser evaluado por el C.A.P. (Consejo Asesor Pedagógico).*”

## 3. Entregas de los avances del Proyecto

### 3.1. Consideraciones

- La presentación del documento integrador de TODOS los avances del proyecto, se entregará en formato digital a través de un formulario oficial para tal fin (<https://entregas.esi.edu.uy>), en los días que aparecen al final del documento, contando hasta las 23:59 h para realizar la entrega, para TODOS los turnos por igual.
- Aquellos grupos que no entreguen en la fecha y horarios estipulados (primera y segunda entrega) tendrán otra oportunidad 3 días pasada la fecha oficial en el horario corrido hasta las 23:59 para TODOS los turnos por igual, pero **este incumplimiento se considerara en forma NEGATIVA en la evaluación de dicha entrega, significando esto una posible calificación negativa.**
- Cada entrega se considerará un documento de avance del proyecto, por lo cual debe incluir la entrega anterior con las correcciones indicadas por el docente, sumado al contenido de cada entrega.
- La documentación integrada para cada uno de los avances deberá ser entregada en formato PDF.

### 3.2. Fechas

- **Conformación de los grupos:** hasta el viernes 30/05
- **Avances y entrega final:**
  - 1ra Entrega: Lunes 27/07
  - 2da Entrega: Lunes 14/09
  - Entrega Final: Viernes 6/11
  - Instalación: del 9/11 al 11/11
- **Defensa final: 16, 17, 18 y 19 de Noviembre**

Los docentes deberán entregar las correcciones indicadas de cada asignatura; en donde el mismo realizará la devolución y/o calificación por escrito a cada grupo de proyecto. Esta devolución no debe superar los 15 días desde la fecha de la entrega.

## 4. Defensa del Proyecto Final

- La Defensa se realiza antes de la reunión final:
- El tiempo de la defensa por grupo se compone de 20 min de exposición y luego 30 minutos para preguntas.
- La misma debe ser tecnológica. Se elimina la idea de que cualquiera debería comprender el contenido de la exposición.
- En esta instancia deben estar presentes todos los docentes del área tecnológica
- La presentación de los estudiantes debe contener información de todas las asignaturas participantes del proyecto. Todos los docentes deben observar cómo se trabajó en las otras asignaturas de manera de tener una visión integrada del proyecto
- Cada grupo decide si admite que su exposición sea escuchada por otras personas (que no sean sus docentes). Se sugiere que la Exposición sea abierta al público, lo que permite al alumno trabajar con público real cuando expone un trabajo.
- Los docentes pueden realizar a los estudiantes las preguntas que consideren pertinentes. Se puede preguntar a todo el grupo o a un alumno en particular. El Tribunal decidirá si durante la Defensa otras personas estarán presentes.
- Quienes escuchan la exposición no pueden preguntar a los alumnos que defienden.
- Luego de finalizada la Defensa, solo los docentes del Tronco Tecnológico presentes le asignan (mediante acuerdo) una calificación a la presentación, la cual se debe agregar a las demás notas que se tienen del año en su asignatura.
- Si un alumno por razones extraordinarias (enfermedad, viaje, u otra) no puede asistir personalmente, puede hacerlo mediante videoconferencia, Skype u otra modalidad.
- Se sugiere que si una asignatura tiene calificación insuficiente en el producto final, la Nota del mismo sea insuficiente. No es lógico que un proyecto integrado sea suficiente cuando una de las partes no lo es.
- Se acepta como proyecto presentar una hoja en blanco (o la carátula) con el nombre del estudiante. El reglamento establece: "La no realización o presentación del proyecto final implica la pérdida del Espacio Curricular Tecnológico", no dice que el mismo debe ser aceptable, por lo que no repite dicho espacio si presenta algo.

## 5. Generalidades del Proyecto

### Sistema de Gestión de Residuos Urbanos – SiGeRU

#### Introducción

El proyecto consiste en el diseño y desarrollo de un sistema de gestión vinculado a la problemática de los residuos urbanos. El software deberá cubrir necesidades básicas de administración, seguimiento y análisis de información relacionada con la recolección, el tratamiento de residuos y el uso de los recursos disponibles. El análisis detallado, el relevamiento de requerimientos y el diseño del sistema quedarán en manos de los estudiantes, aplicando las técnicas aprendidas en clase. Este documento establece un marco de **funcionalidades mínimas y opcionales**, que servirá como guía para orientar el trabajo.

#### Áreas principales del sistema

##### 1. Gestión de información

El sistema debe permitir administrar los elementos centrales vinculados a la gestión de residuos:

- **Contenedores:** alta, baja y modificación; ubicación; estado (funcional, roto, desbordado); tipo de residuos.
- **Usuarios y roles:** administrador municipal, cuadrilla de recolección, operarios de vertederos/centros de acopio, vecinos (opcional).
- **Centros de acopio y vertederos:** registro de instalaciones, capacidad, tipo de residuos que gestionan.
- **Rutas de recolección:** posibilidad de registrar y asignar rutas a cuadrillas (mínimo a nivel conceptual).

##### 2. Gestión de incidencias

Módulo para registrar y dar seguimiento a problemas o eventos relevantes:

- Incidencias en contenedores (rotura, desborde, falta de recolección).
- Estado de cada incidencia: abierta, en curso, resuelta.
- Asignación de cuadrillas responsables.
- Historial de incidencias por zona o contenedor.
- Notificaciones/reclamos de vecinos (opcional).

##### 3. Gestión de recursos

Administración de los medios materiales utilizados en la recolección y el procesamiento:

- **Nivel básico:** inventario de camiones recolectores (alta/baja, estado) e inventario simple de contenedores de repuesto.
- **Nivel intermedio:** gestión detallada de la flota de camiones (asignación de rutas, mantenimiento preventivo, disponibilidad), maquinaria básica en vertederos y centros de acopio.

- **Nivel avanzado:** registro de logística completa (traslado de residuos, reparación de contenedores, herramientas y equipamiento de plantas de procesamiento), con informes de disponibilidad y uso.

## 4. Informes y reportes

El sistema debe ofrecer reportes que apoyen la toma de decisiones:

- **Básicos:** listado de incidencias abiertas y cerradas, volumen de residuos recolectados (con datos simulados).
- **Estadísticas:** zonas críticas, tiempos de resolución, frecuencia de recolección.
- **Panel de control:** visualización gráfica de datos relevantes.
- **Mapa interactivo:** ubicación de contenedores, incidencias, centros de acopio y recursos (camiones).
- **Avanzados:** informes de capacidad de vertederos, clasificación y reciclaje de residuos, proyecciones de llenado de contenedores o sobrecarga en determinadas zonas.

## Alcances sugeridos

1. **Nivel mínimo (obligatorio):**
  - 1.1. ABM de usuarios, contenedores y camiones.
  - 1.2. Registro y seguimiento de incidencias.
  - 1.3. Inventario básico de camiones y contenedores de repuesto.
  - 1.4. Informe simple (ej. listado de incidencias abiertas/cerradas).
2. **Nivel intermedio (recomendado):**
  - 2.1. Panel de control con estadísticas.
  - 2.2. Mapa interactivo con contenedores e incidencias.
  - 2.3. Gestión de flota (estado, mantenimiento, asignación).
  - 2.4. Registro de centros de acopio y maquinaria básica.
3. **Nivel avanzado (opcional):**
  - 3.1. Sistema de notificaciones o alertas.
  - 3.2. Gestión avanzada de logística (reparación, traslado, maquinaria especializada).
  - 3.3. Módulos de reciclaje y clasificación de residuos.
  - 3.4. Predicciones de llenado y generación de residuos por zona.

## Conclusión

El proyecto fija un marco claro de necesidades a cubrir y define distintos niveles de alcance para asegurar que todos los grupos logren un producto funcional. A la vez, brinda libertad para que cada grupo realice su relevamiento, análisis y diseño propio, aplicando su creatividad y conocimientos para alcanzar un resultado significativo y de impacto en la comunidad.

## 6. Detalle de Implementación

El sistema se compone de varias aplicaciones interactuando entre sí, implementando una arquitectura distribuida de microservicios. Tanto la landing page, la aplicación de los usuario, como las APIs que componen el Backend deben ser accesibles por el usuario a través de internet.

### 6.1. Landing Page

Esta aplicación tiene como objetivo brindar información al público general, así como recibir solicitudes de registro de usuario.

**Esta aplicación no accede directamente a ningún tipo de información ni la almacena. Debe interactuar con su API correspondiente**

### 6.2. Frontend de Usuarios

Esta aplicación constituye la interfaz de interacción para el usuario final. Es responsable de gestionar el registro, el inicio de sesión y la visualización de los perfiles, adaptando la experiencia según el rol asignado (Administrador, Cuadrilla, Operario o Vecino).

Para garantizar la seguridad y el desacoplamiento, el frontend tiene restringidas las operaciones según los permisos del usuario autenticado. Esta aplicación no almacena datos localmente ni accede de forma directa a la base de datos; todas sus funciones se ejecutan mediante el consumo de las APIs correspondientes, asegurando que la lógica de negocio y la persistencia residan exclusivamente en el backend.

### 6.3. API de Usuarios

Esta aplicación es una API RESTful encargada del registro, la autenticación y la gestión de perfiles. Se ocupa de recibir nuevas solicitudes de registro para su posterior aprobación, gestionar el inicio de sesión desde los front ends y administrar la información personal y los roles (Administrador, Cuadrilla, Operario, Vecino, etc.).

### 6.4. API de Gestión

Esta aplicación es una API RESTful que actúa como el orquestador operativo del sistema. Se encarga de la gestión de los elementos físicos fijos y el inventario: alta/baja de contenedores, estado de los centros de acopio y vertederos (capacidad, tipo de residuos) y el stock de repuestos.

### 6.5. API de Recolección

Esta aplicación es una API RESTful dedicada a la logística móvil. Se encarga de la gestión de la flota de camiones (mantenimiento y estado), la asignación de rutas para las cuadrillas y el seguimiento del inventario de recursos utilizados durante la ruta de recolección.

## 6.6. Backoffice de Administración

Esta aplicación implementa un panel de control centralizado diseñado para la gestión municipal. Su función principal es administrar los parámetros críticos del sistema, incluyendo la gestión de usuarios y roles, el monitoreo de incidencias en tiempo real y la visualización de mapas interactivos con la ubicación de contenedores y recursos. Asimismo, el módulo permite la generación y exportación de reportes estadísticos detallados sobre la recolección, fundamentales para la toma de decisiones operativas.

Este sistema interactúa con la base de datos de forma directa para tareas de administración masiva, aunque puede consumir servicios de las APIs para validaciones específicas. La implementación queda a criterio de los alumnos, pudiendo desarrollarse de forma monolítica o separando Frontend y API.

### Arquitectura del Sistema

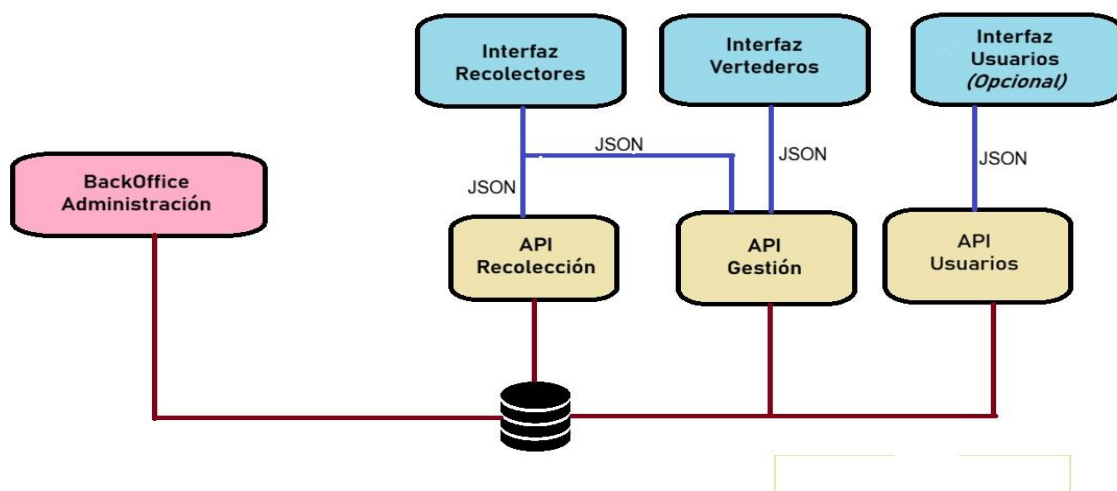
El sistema debe estar desarrollado siguiendo parámetros actuales de desarrollo de software.

Cada aplicación compone una base de código distinta e independiente. Además, cada aplicación debe implementar la arquitectura MVC.

Los Frontend del sistema solo deben utilizar HTML, CSS para su diseño, y JavaScript para las interacciones con el usuario y llamadas a las APIs correspondientes.

Las APIs deben desarrollarse e implementarse utilizando PHP, implementando el estándar de API REST, además seguir los lineamientos de un código limpio y mantenible..

A continuación se detalla visualmente la interacción de las diferentes aplicaciones:



## 7. Requisitos generales

- El grupo deberá constituir una empresa, configurando teóricamente PYMES, para lo cual deberán :
  - Definir su Misión y Visión.
  - Diseñar un Plan de Negocios y los Comprobantes necesarios y convenientes para la operativa de la empresa.
  - Especificar los trámites que deberán realizar ante los organismos oficiales a los efectos de tener la empresa en regla.
  - Estudiar la viabilidad del negocio.
- Toda la propuesta deberá estar enmarcada en la tecnología de Gestión de Proyectos, debiendo presentarse cronograma de ejecución y detalle de las etapas constitutivas del mismo.
- El software deberá estar debidamente documentado, atendiendo a algunos de los modelos de análisis estudiados durante el curso. **Cada docente especificará qué mínimos son deseables alcanzar en el cumplimiento del proyecto.**
- El software deberá ser flexible, permitiendo parametrizar datos.
- Dado a la importancia de un desarrollo de software coherente y consistente, se deben implementar diversos tipos de pruebas en todo el proceso de desarrollo de las aplicaciones, en todo nivel correspondiente (Unit Tests, Feature Tests, Usability Tests, por ejemplo).
- El idioma del software debe ser parametrizable.
- Se debe tener en cuenta la seguridad informática tanto a nivel de software como a nivel de telecomunicaciones. Elaborar estrategias de seguridad en cada capa, complementaria una de otra.
- Es tarea de los estudiantes investigar e implementar detalles no explícitos en la letra, trabajando así conjuntamente con los docentes y apelar a la calidad del producto.
- También, todo el código generado debe versionar con la herramienta Git, y entregarse en un repositorio de Código hosteado en el servicio GitHub (<https://github.com>).
- El sistema debe cumplir con TODAS las características indicadas anteriormente
- En caso de agrandar el alcance del proyecto determinar siempre en la big picture y en las diferentes documentaciones. Se valoran todas aquellas funcionalidades, características y performance del sistema que se desarrolle **luego de finalizados los requerimientos de la letra.**

## 8. Herramientas y ambiente

- **Frontend:** HTML5, CSS y JavaScript.
- **Backend:** PHP versión 8.
- **Almacenamiento:** MySQL 8 como Motor de Base de Datos, instalado en el servidor de base de datos de la Escuela, con sistema operativo.
- **Ambiente del servidor Web:** Linux, ejecutando cada aplicación desarrollada en contenedores Docker.
- **Código fuente** de todas las asignaturas versionado con la herramienta Git, y hosteado en el servicio GitHub.
- **El incumplimiento de los puntos anteriores generará una penalización en la calificación.**

## 9. Requerimientos por Unidad Curricular

La lista de requerimientos que se indica a continuación, es tentativa y no exhaustiva; cada docente indicará qué puntos serán relevantes para él, pudiendo agregar a la misma, aquellos puntos que no estuvieran presentes, o quitar de la misma, aquellos que no se consideren oportunos.

**Estos cambios deberán ser informados a la Coordinación de Informática y a la Dirección de la Escuela, acompañados por una justificación, y serán aprobados por ambas partes. En caso de aprobarse, deberá notificarse por correo electrónico a los alumnos del grupo involucrado.**

Los requerimos de las unidades curriculares que faltan se irán por parte de los docentes en la medida en que se vayan definiendo, además de agregarse a este documento de ser pertinente.

### 9.1. Ingeniería de Software

#### 1er Entrega

- Para este proyecto se propone la siguiente metodología:
  - Fundamentación del modelo de desarrollo a seguir.
- Pre-Análisis y Análisis:
  - Relevamiento de Datos.
    - Empleo de técnicas vistas durante el curso.
    - Elaboración de formularios apropiados para la tarea anterior.
  - Especificación de Requerimientos (Estándar IEEE830).
- Diagrama Entidad Relación, con RNE
- Pasaje a tablas normalizado (3ra. Forma Normal).

## 2da Entrega

- **Análisis:**
  - Cálculo de métricas: Análisis por punto de función (costo, esfuerzo, duración).
  - Estudio de factibilidades.
    - Operativos, económicos, técnicos y legales.
  - Análisis Costo-Beneficio.
  - Gantt:
    - Primera versión con línea base
    - Segunda versión con los Sprints
- **Diseño:**
  - Diagrama de clases.
  - Modelo diagrama Casos de uso (planilla y diagrama).
  - Diagrama de Paquetes.
  - Diagrama de Estados (opcional)

## 3er Entrega

- **Testing y Finalización del Producto:**
  - Plan de testing.
  - Justificación de Caja Blanca
  - Caja Negra, de interfaz entre otras.
  - Casos de Prueba con juegos de datos.
  - Manuales de Uso por perfiles de usuario.
  - Manual de Instalación del Sistema.
  - Manuales de Administración del Sistema.
  - Videos Tutoriales (Opcional).

## 9.2. Proyecto UTU LAB

### 1er Entrega

- **Planteo del desafío real:**
  - Breve presentación del problema que aborda el proyecto (basado en la consigna).
  - Fundamentación de la relevancia social del mismo.
- **Aplicación de herramientas de investigación (CET2 - CL1):**
  - Aplicar al menos una herramienta analítica para relevar información sobre el contexto y usuarios:
    - Ejemplos: Observación directa, entrevistas, mapeo, análisis FODA, estudio de casos.

- Construcción de una persona (usuario del sistema):
  - Diseño de al menos una persona (usuario ficticio representativo) que utilice el sistema.
- Etapas E1 y E2 del Pensamiento de Diseño (CET2 - CL2 y CL3):
  - E1: Exploración del contexto, del problema y del usuario.
  - E2: Definición clara del problema y de los objetivos.
  - Presentación en pizarra colaborativa (se sugiere Jira o Trello).
- Aplicación de una herramienta de creatividad (CET2 - CL3):
  - Elegir una para generar ideas de solución: SCAMPER, brainstorming, mapa mental, método 635.
- Primera versión de idea/propuesta de solución:
  - Descripción preliminar de cómo el sistema va a dar respuesta al problema.
- Primera maqueta analógica o digital (CET1 - CL1):
  - Esbozo o maqueta simple (papel, cartón, software) del flujo del sistema o de su aspecto visual.
- Registro de proceso de trabajo y colaboración:
  - Breve retrospectiva del equipo: ¿Qué funcionó bien?, ¿Qué se puede mejorar?

## 2da Entrega

1. Prototipo funcional lo-fi del sistema (CET1 - CL1):
  - Crear una versión inicial del sistema con herramientas analógicas o digitales (puede ser Figma, papel, cartón, dibujos funcionales).
  - Debe representar la experiencia del usuario como cooperativista o administrativo.
2. Aplicación de herramienta de validación (CET2 - CL4):
  - Validación con usuarios reales o simulados (pueden ser docentes, compañeros u otros usuarios).
  - Registro del feedback recibido y principales aprendizajes.
3. Revisión de la definición del problema:
  - ¿Cambia o se ajusta algo en base a lo descubierto?
4. Ajustes en la propuesta de solución:
  - Nuevas ideas o decisiones tomadas luego del testeo del prototipo.
5. Consideraciones sobre materiales y sostenibilidad (CET1 - CL1):
  - Reflexión sobre los materiales elegidos y sus implicancias ambientales y de durabilidad.
6. Registro del proceso de iteración:
  - Evidencias del trabajo (fotos, capturas, documentos).
  - Segunda retrospectiva: mejoras en el trabajo en equipo, dificultades superadas.

## 3er Entrega

- Prototipo high-fidelity o maqueta avanzada (CET1 - CL1 y CL4):
  - Modelo del sistema terminado o muy cercano al resultado final.
  - Puede ser digital (Figma, simulador) o físico si corresponde al aspecto visual.
- Aplicación combinada de tecnologías (CET1 - CL1):

- Incluir alguna fusión entre herramientas analógicas y digitales (diseño en papel + simulación digital, o maqueta + pantalla interactiva, etc.).
- Documento de validación final (CET2 - CL4):
  - Nueva instancia de testing con usuarios.
  - Comparación entre la versión inicial y la final: mejoras y decisiones tomadas.
- Informe reflexivo del proceso completo:
  - ¿Qué aprendió el grupo?
  - ¿Qué errores cometieron y cómo los solucionaron?
  - ¿Qué impacto tiene este sistema para su comunidad?
- Documentación visual del proyecto:
  - Big Picture (diagrama visual general del sistema)
  - Línea de tiempo del desarrollo (con hitos principales)
  - Capturas de prototipos y entrevistas
- Preparación para defensa del proyecto:
  - Presentación clara de las decisiones de diseño tomadas desde UTULAB.
  - Participación activa de todos los integrantes.

## 9.3. Administración de Sistemas Operativos

Para Administración de Sistemas Operativos, se deberá contemplar que la implementación final deberá contar con 3 servidores como mínimo:

- Un servidor donde se ejecutarán las aplicaciones desarrolladas en Programación Full Stack
- Un servidor donde se ejecutará la Base de Datos MySQL
- Un servidor donde se almacenarán los respaldos de la información.

### 1er Entrega

- Relevamiento del S.O. para el servidor.
- Manual de instalación del S.O. servidor.
- Instalación y configuración de la red.
- Gestión y administración de paquetes del S.O. elegido.

### 2da Entrega

- Configuración del servicio SSH en el servidor.
- Definición de medios de respaldo a largo plazo y alta disponibilidad.
- Rutinas de backup y scripts correspondientes.
- Primera versión del script del servidor.

### 3er Entrega

- Script final del servidor.
- Gestión y administración de contenedores para despliegue de aplicaciones.
- Generación del servidor de respaldo.

**Logros mínimos necesarios para administración de sistemas operativos:**

Realiza un manual de instalación para el servidor Linux aplicado en el proyecto con soporte técnico empresarial (SUSE Linux Enterprise Server, Red Hat Enterprise Linux, CentOS, Ubuntu Server) para el usuario final.

Configura la red en el servidor Gnu/Linux de manera estática para ser visible fácilmente desde la red. Configura los puertos del servidor para enviar información a través de la red.

Configura el servicio SSH en el servidor ajustado a los requerimientos para ser utilizado para realizar conexiones de manera remota.

Realizar una primera versión del script de administración del servidor donde se contemplan los scripts trabajados en clase. Desarrolla el script de manera modular en partes pequeñas y reutilizables, lo que facilita su mantenimiento, depuración y escalabilidad.

Implementa el script final del servidor de manera modular con todos los script necesarios vistos en el curso:

- Gestión de usuarios
- Gestión de grupos
- Gestión de respaldos
- Gestión de redes
- Gestión de Bases de datos
- Gestión de Firewall
- Gestión de Logs del Sistema

Para poder administrar de manera eficiente todo el servidor.

## 9.4. Programación Full Stack

### 1er Entrega

- Frontend:
  - Landing Page (página principal).
  - Capacidad de registro de Usuarios nuevos y login (solo navegabilidad).
  - Listados y formularios de usuarios, contenedores y camiones.
- Backend:
  - APIs básica de usuarios y gestión.
  - Inicio de sesión.
  - Registro de usuario, contenedores y camiones.
  - Endpoints para obtener usuarios, contenedores y camiones.
  - Todo a nivel de código/sesión, sin persistencia de datos con BD.
- Github
  - Creación de una organización en GitHub
  - Código versionado utilizando Git, con sus correspondientes repositorios, y un correcto historial de cambios.
- Base de datos
  - Primera versión del modelo físico de la BD.
  - DDL de la creación de la base de datos y sus tablas
  - Dump de la estructura de la BD

## 2da Entrega

- Frontend:
  - Correcciones de la primera entrega.
  - Registro e inicio de sesión completo.
  - CRUD completos de usuarios (todos los roles).
  - CRUD de contenedores.
  - CRUD de camiones (flota).
  - CRUD de centros de acopio y maquinaria básica.
- Backend:
  - Endpoints para los CRUDs mencionados.
  - Persistencia de datos con DB.
  - Seguridad y separación de endpoints por roles.
- Github
  - Código versionado utilizando Git, con sus correspondientes repositorios, y un correcto historial de cambios.
- Testing
  - Funcionamiento en la API, mediante código (test unitarios).
- Archivos necesarios para ejecutar las aplicaciones utilizando Docker
- Base de datos
  - Correcciones de la primera entrega.
  - Datos de prueba cargados en la base de datos.

## 3er Entrega

- Implementación final de Frontend y Backend.
  - Correcciones de las anteriores entregas.
  - API recolección.
  - Registro y seguimiento de incidencias.
  - Mapa interactivo con contenedores e incidencias.
  - Informes simples (ej. listado de incidencias abiertas/cerradas).
  - Panel de control con estadísticas (ej. top de contenedores desbordados del mes).
  - Sistema de notificaciones o alertas.
- Github
  - Código versionado utilizando Git, con sus correspondientes repositorios, y un correcto historial de cambios.
- Testing
  - Cobertura de CRUDs y funcionalidades principales del sistema mediante código (tests unitarios).
- Archivos necesarios para ejecutar las aplicaciones utilizando Docker.
- Base de datos
  - Correcciones de las anteriores entregas.

## 9.5. Emprendedurismo y Gestión

### 1er Entrega: Creación Empresa y Estrategia Corporativa

#### 1. CARÁTULA:

- Logo del CETP-UTU y
- Logo del instituto educativo (escuela) Logo y nombre de la empresa
- Integrantes del equipo ( C.I) Curso
- Asignatura Docente

#### 2. ÍNDICE

#### 3. RESUMEN EJECUTIVO: (Corresponde a la tercera entrega)

- Breve descripción del proyecto que tiene como objetivo captar el interés de los posibles inversores, bancos, gobierno, o lectores; asegurándose de que éstos profundicen en el documento para lograr su participación y/o evaluación positiva.
  - Nombre de la empresa
  - Determinación de la actividad y giro de la misma.
  - Estudio y fundamentación de la opción.
  - Proceso de selección del producto o servicio

#### 4. LOGO:

- Diseño y fundamentación.
- Gráfico identificador de la empresa.

#### 5. PRESENTACIÓN:

- Visión, Misión, Objetivos y Valores de la organización.

#### 6. FORMA JURÍDICA:

- Elección de la forma jurídica, justificación de la opción. Giro o actividad. Reglamentaciones vinculadas a la actividad.

#### 7. EMPLAZAMIENTO:

- Ubicación geográfica, justificación.
- Plano de la ubicación de la empresa y de la competencia. Acceso de clientes, estacionamiento y visibilidad.
- Fotografías de la locación.
- Precio de Venta o del Alquiler del local u oficina.

#### 8. HERRAMIENTA:

- Diagrama de Gantt del plan de negocio.

## 2da Entrega: Estudio de Mercado, Planes de Inversiones y Recursos Humanos

### 9. ANÁLISIS DEL SECTOR:

- Breve resumen de la situación actual del sector en que se desarrollará la empresa, (estructura: monopolio, oligopolio o competitivo, etc.)

### 10. ANÁLISIS DEL ENTORNO:

- Descripción general del mercado. (extensión geográfica, tamaño y clientes potenciales).
- Competidores,
- Proveedores,
- Organismos, etc.

### 11. MATRIZ FODA:

- Determinar las fortalezas, debilidades, amenazas y oportunidades de la empresa, tener en cuenta:
- Factores económicos: crecimiento, tendencias, sistema impositivo, tendencia de precios de insumos.
- Factores legales, gubernamentales, políticos y medioambientales.
- Otros factores independientes al control de la empresa.
- A partir de la matriz establecer las estrategias a utilizar.

## 12. PLAN DE INVERSIONES

### ● RECURSOS MATERIALES

- Recursos necesarios para la puesta en marcha.
- **Activos corrientes: Inventarios**
- Determinación de la cantidad de bienes de cambio que serán necesarios (mercaderías o materias primas)
- Determinación de la forma de control y venta.

### ● **Activos no corrientes:**

- Planta y equipamiento (maquinaria herramientas equipos), Construcciones (terrenos, edificios, refacciones, ampliaciones Determinación y descripción de los bienes de uso necesarios.

- Presupuesto Total en Recursos Materiales.

### ● **PLAN DE RECURSOS TECNOLÓGICOS**

Tecnología a aplicar en el emprendimiento.

- Presupuesto Total en Recursos Tecnológicos.

## 13. PLAN DE RECURSOS HUMANOS:

- Organigrama - estructura de la empresa.
- Determinación de los cargos, funciones y tareas a desempeñar para el cumplimiento de los objetivos. Personal necesario para el funcionamiento de la empresa (cantidad).
- Normativa Laboral relacionada con el sector Informático.
- Diseño de recibos de sueldos.
- Compensaciones o beneficios como forma de motivación.
- Presupuesto total en Recursos Humanos.

## 3er Entrega: Planes de Comercialización y Económico - Financiero

### PLAN DE COMERCIALIZACIÓN:

- Identificación del consumidor, segmentos del mercado.

- Presentar una encuesta y sus conclusiones.
- Análisis de la Competencia como se divide el mercado.
- Determinación de la estrategia de ingreso al mercado - posicionamiento.
- Análisis: ¿Por qué será rentable ? y ¿Oportunidades de crecimiento?
  
- Descripción del producto o servicio que se ofrece, descripción de los beneficios desde el punto de vista del cliente ¿qué vende?, establecimiento de la ventaja competitiva, diferenciación de los competidores.
  
- Formas de distribución, forma de determinación del precio de venta. Diferentes tipos de promoción a utilizar y su justificación.
  
- Publicidad: Indicar: ¿Dónde? ¿Cuándo? y ¿Cómo se dará a conocer el emprendimiento y sus productos o servicios?. ¿Durante cuánto tiempo y cuál será su costo?

#### **PLAN RECURSOS FINANCIEROS:**

- Estimación de ingresos (precio x cantidades producidas en x cantidad de años).
  
- Estimación de gastos ( Costos fijos y costos variables en qué incurre la empresa. Gastos administrativos funcionamiento, sueldo del personal administrativo y/o producción, gastos en material de limpieza. Gastos de papelería, impuestos etc.).
  
- Presupuesto: realizar un presupuesto (proyección de ingresos y gastos). Flujo de Caja. Determinar el Capital del GIRO (monto necesario para iniciar el negocio).
  
- Solicitud de préstamo para financiar la inversión inicial, justificar utilización, forma de financiamiento plazo y garantía.
  - Determinar el Punto de Equilibrio.
  
  - Determinar el Margen de Utilidad.

- Viabilidad de la inversión: Valor Actual Neto (VAN) y Tasa Interna de Retorno (TIR).

#### **16. ANEXOS:**

- Contrato Societario
- Trámites al inicio de la empresa:
  1. Presentar formularios de apertura en DGI y BPS.
  2. Presentar tarjeta de RUT.
  3. Presentar Constancia de impresión.
  4. Presentar planilla de trabajo.
  5. Presentar formularios de IMM y BSE.
  6. Presentar todo formulario de diferentes organismos que tengan relación con la actividad que realiza la empresa
  7. Comprobantes exigidos por la ley.

Nota: La presente guía podrá ser adaptada por cada docente de la asignatura en función de sus decisiones pedagógicas.

## 10. RÚBRICAS

### 10.1. Ingeniería de Software

#### Rúbrica de evaluación para primera entrega

Criterio	Insuficiente (1-2)	Escaso (3-4)	Moderado (5-6)	Significativo (7-8)	Destacado (9-10)
Metodología y Relevamiento	No presenta fundamentación ni datos.	Datos incompletos y técnicas mal aplicadas.	Presenta fundamentación básica y formularios de relevamiento funcionales, aunque con omisiones o errores menores de formato.	Relevamiento detallado o con análisis de técnicas aplicadas.	Justificación metodológica profunda y relevamiento exhaustivo.
Req. (IEEE830)	Documento ausente.	Requerimientos vagos o mal clasificados.	Lista de requerimientos funcionales y no funcionales de forma clara, se aceptan imprecisiones leves en la redacción técnica.	Especificación precisa bajo estándar IEEE830.	Documentación profesional, trazable y sin ambigüedades.

Modelo de Datos	Sin MER ni tablas.	MER con errores de lógica y sin normalizar.	MER completo con RNE y pasaje a tablas en 3FN, se aceptan errores menores de sintaxis o normalización que no afecten la lógica base.	Modelo de datos sólido y normalizado y sin errores técnicos.	El modelo relacional está optimizado para el rendimiento y escalabilidad.
-----------------	--------------------	---	--	--	---

### Rúbrica de evaluación para segunda entrega

Criterio	Insuficiente (1-2)	Escaso (3-4)	Moderado (5-6)	Significativo (7-8)	Destacado (9-10)
Métricas y Factibilidad	Sin estudios de costo o esfuerzo.	Cálculos erróneos o factibilidad incompleta.	Cálculo básico de Puntos de Función y estudio de factibilidad (OTEL), se aceptan desvíos menores en las estimaciones o costos.	Análisis Costo-Beneficio detallado y métricas coherentes.	Análisis financiero y operativo riguroso con proyecciones reales.
Planificación (Gantt)	No hay cronograma.	Gantt desactualizado o sin hitos.	Gantt con línea base y organización por Sprints.	Cronograma realista con asignación de recursos.	Planificación ágil óptima con control de desviaciones.
Diseño del Sistema	Sin diagramas de clases o casos de uso.	Diagramas inconsistentes o con errores de sintaxis UML.	Diagrama de clases, casos de uso (planilla/diagrama) y paquetes básicos, se aceptan errores menores de	Diseño integral con correcta relación entre componentes.	Modelado UML avanzado, incluyendo estados y diseño modular.

			notación UML.		
--	--	--	---------------	--	--

### Rúbrica de evaluación para tercera entrega

Criterio	Insuficiente (1-2)	Escaso (3-4)	Moderado (5-6)	Significativo (7-8)	Destacado (9-10)
Testing	No se realizaron pruebas.	Casos de prueba sin datos o mal estructurados.	Plan de testing con casos de Caja Blanca y Caja Negra básicos, se aceptan juegos de datos simples o errores menores de cobertura.	Testing detallado con juegos de datos reales y resultados verificados sin errores..	Estrategia de testing exhaustiva con corrección de errores documentada.
Documentación Técnica	Ausencia de manuales.	Manuales incompletos o difíciles de seguir.	Manuales de Uso (perfiles), Instalación y Administración funcionales, se aceptan errores de menores o capturas desactualizados.	Manuales claros, bien ilustrados y técnicamente correctos.	Documentación profesional completa, incluyendo soporte multimedia.

## 10.2. Proyecto UTU Lab

### Requerimientos de Proyecto UTULab 1er Entrega

- Planteo del desafío real:
- Breve presentación del problema que aborda el proyecto (basado en la consigna).
- Fundamentación del impacto social del mismo.
- Reglamento interno de la empresa.
- Aplicar al menos una herramienta analítica para relevar información sobre el contexto y usuarios:
  - Ejemplos: Observación directa, entrevistas, mapeo, análisis FODA, estudio de casos.
- Diseño de al menos una persona (usuario ficticio representativo) que utilice el sistema.
- Etapas E1 y E2 del Pensamiento de Diseño:
  - E1: Exploración del contexto, del problema y del usuario.
  - E2: Definición clara del problema y de los objetivos.
- Presentación en pizarra colaborativa (se sugiere Jira o Trello).
- Aplicación de una herramienta de creatividad:
  - SCAMPER, brainstorming, mapa mental, método 635.
- Primera versión de idea/propuesta de solución
- Descripción preliminar de cómo el sistema va a dar respuesta al problema.
- Primera versión de prototipo Low-Fi (enfocado en navegabilidad).
- Registro de proceso de trabajo y colaboración
- Breve retrospectiva del equipo: ¿Qué funcionó bien?, ¿Qué se puede mejorar?

### 2da Entrega

- Prototipo funcional lo-fi del sistema
- Aplicación de herramienta de validación:
  - Validación con usuarios reales o simulados (pueden ser docentes, compañeros u otros usuarios).
  - Registro del feedback recibido y principales aprendizajes.
- Ajustes en la propuesta de solución
- Registro del proceso de iteración
- Segunda retrospectiva: mejoras en el trabajo en equipo, dificultades superadas.

### 3er Entrega

- Prototipo high-fidelity o maqueta avanzada:
- Documento de validación final:
- Informe de testing con usuarios.
- Comparación entre la versión inicial y la final: mejoras y decisiones tomadas.
- Informe reflexivo del proceso completo:
  - ¿Qué aprendió el grupo?
  - ¿Qué errores cometieron y cómo los solucionaron?

- ¿Qué impacto tiene este sistema para su comunidad?
- Documentación visual del proyecto:
  - Big Picture (diagrama visual general del sistema)
- Línea de tiempo del desarrollo (con hitos principales)
- Capturas evolutivas de prototipos

## 10.3. Administración de Sistemas Operativos

### Rúbrica de evaluación para primera entrega

Criterios de evaluación	1-3 puntos	4-6 puntos	7-9 puntos	10 puntos
Relevamiento del S.O. para el servidor	No realiza este punto para la entrega	Realiza un relevamiento mínimo de los sistemas operativos eligiendo uno para ser utilizado en el servidor.	Realiza un relevamiento y justifica el Sistema Operativo a utilizar en el servidor, no incluye plan de soporte técnico claro para orientar al usuario final.	Realiza un relevamiento exhaustivo y justifica el Sistema Operativo a utilizar en el servidor, incluyendo planes de soporte técnico para el usuario final.
Manual de instalación del S.O. servidor	No realiza este punto para la entrega	Realiza un manual básico de instalación con capturas de pantallas no tan claras para el usuario, no se contemplan todas las partes de la instalación ni se explica el soporte	Realiza un manual de instalación básico con capturas claras para el usuario final, no brinda manual de soporte.	Realiza un manual de instalación para el servidor Linux aplicado en el proyecto con soporte técnico empresarial (SUSE Linux Enterprise Server, Red Hat Enterprise Linux, CentOS, Ubuntu Server) para el usuario final.

Instalación y configuración de la red	No realiza este punto para la entrega	Instala el servidor pero deja la red en formato DHCP para ser configurado más adelante	Instala el servidor y configura la red, pero no contempla los puertos necesarios que se deben de modificar para ser visible por la red en base al proyecto.	Configura la red en el servidor Gnu/Linux de manera estática para ser visible fácilmente desde la red. Configura los puertos del servidor para enviar información a través de la red
Gestión y administración de paquetes del S.O.	No realiza este punto para la entrega	Instala solo los paquetes por defecto en el sistema operativo.	Instala y configura los paquetes básicos del sistema operativo, no logra configurar el sistema para realizar un despliegue de la aplicación	Instala y configura de manera eficiente los paquetes necesarios para que el servidor pueda ser utilizado para hacer un despliegue inicial de la aplicación

### Rúbrica de evaluación para segunda entrega

Criterios de evaluación	1-3 puntos	4-6 puntos	7-9 puntos	10 puntos
Configuración del servicio SSH	No realiza este punto para la entrega	Instala el servicio SSH en el servidor pero no aplica las configuraciones básicas	Instala el servicio SSH cliente/servidor y configura los puertos básicos	Configura el servicio SSH en el servidor ajustado a los requerimientos, siendo utilizado para realizar conexiones de manera remota generando llaves públicas distribuidas en las conexiones

Definición de respaldos	No realiza este punto para la entrega	Define de manera teórica los respaldos que serán realizados en el servidor	Define de manera teórica los respaldos e implementa de manera básica los script de automatización.	Define medios de respaldo a largo plazo y alta disponibilidad de los datos para evitar pérdidas de información
Rutinas de Backups y Scripts básicos	No realiza este punto para la entrega	Define de manera teórica las rutinas de backups pero no implementa de manera correcta los scripts de automatización	Define de manera teórica las rutinas de backups y implementa scripts básicos de automatización	Programa rutinas de backup, y sus correspondientes scripts, configurados de manera automática para minimizar pérdidas de datos.
Primer script de administración del servidor en formato modular.	No realiza este punto para la entrega	Implementa una base del script del servidor pero no cumple con los criterios de modularización	Implementa varios scripts para administrar el servidor pero no logra realizar una correcta conexión entre ellos.	Realizar una primera versión del script de administración del servidor donde se contemplan los scripts trabajados en clase. Desarrolla el script de manera modular en partes pequeñas y reutilizables, lo que facilita su mantenimiento, depuración y escalabilidad

### Rúbrica de evaluación para tercera entrega

Criterios de evaluación	1-3 puntos	4-6 puntos	7-9 puntos	10 puntos
Script final del servidor	No realiza este punto para la entrega	Implementa el script final del servidor con la gestión de usuarios y grupos completa	Implementa el script final del servidor con la gestión de usuarios, grupos y redes completa, quedan parcialmente completos el resto de los scripts	Implementa el script final del servidor de manera modular con todos los script necesarios vistos en el curso: <b>Gestión de usuarios</b> <b>Gestión de grupos</b> <b>Gestión de respaldos</b> <b>Gestión de redes</b> <b>Gestión de Bases de datos</b> <b>Gestión de Firewall</b> <b>Gestión de Logs del Sistema</b> Para poder administrar de manera eficiente todo lo relacionado a la infraestructura.
Gestión y administración de contenedores para despliegue de aplicaciones	No realiza este punto para la entrega	Realiza un despliegue básico de la aplicación en un solo contenedor, no maneja volúmenes de datos	Realiza un despliegue parcial en dos contenedores donde se contempla la aplicación y el volumen de datos, no se gestiona la red de manera eficiente.	Gestiona y administra los contenedores del proyecto usando la herramienta docker-compose para hacer un despliegue de la aplicación del proyecto de manera eficiente.

Generación del servidor de respaldo	No realiza este punto para la entrega	Realiza el manual de instalación del servidor de respaldos pero no lo implementa	Realiza el manual y la instalación del servidor de respaldo pero no lo configura	Generará un servidor de respaldo de datos que permite realizar un backup íntegro del sistema como sistema de contingencia ante pérdidas importantes.
-------------------------------------	---------------------------------------	--	--	--

## 10.4. Programación Full Stack

## 10.5. Emprendedurismo y Gestión

Contenidos	Criterios de Logros	Nivel de Avance
	La entrega está completa con todo lo solicitado y es de alta calidad . Muestra calidad de contenido, calidad estética, y está debidamente documentada..	Avance Destacado 9 -10
	La entrega contiene todos los puntos solicitados pero requiere ajustes. Faltan algunos componentes.	Avance Significativo 7-8
	La entrega cumple con lo mínimo aceptable. Demuestra trabajo pero requiere ajustes.	Avance Moderado 5-6
	La entrega está incompleta, o falta contenido	Avance Escaso 3-4
	Faltan elementos importantes. El Contenido está incompleto. No existe contenido y/o no es lo solicitado y requiere cambios profundos.	Avance Mínimo 1-2

11. **Anexo Cartas** (Las cartas deberán ser enviadas a [esi.buceocoordinacion@gmail.com](mailto:esi.buceocoordinacion@gmail.com) y al correo del profesor de "ProyectoUTUlab" desde el correo del alumno que solicita).

**Modelo de carta de abandono de Proyecto:**

Montevideo, \_\_\_\_\_

**ABANDONO DE PROYECTO**

Atte. Coordinador de ESI Buceo  
Apellidos, Nombre: \_\_\_\_\_  
CI: \_\_\_\_\_  
Grupo actual de clase: \_\_\_\_\_  
Grupo actual de proyecto: \_\_\_\_\_

A través de la siguiente, presento solicitud de abandono de la realización del proyecto y a su grupo de trabajo por este año teniendo en cuenta que para poder realizarlo en otro momento deberé abandonar también alguna asignatura del tronco tecnológico.

\_\_\_\_\_  
FIRMA

**Modelo de carta de renuncia a nota de proyecto de años anteriores:**

Montevideo, \_\_\_\_\_

**RENUNCIA A NOTA DE PROYECTO**

Atte. Coordinador de ESI Buceo  
Apellidos, Nombre: \_\_\_\_\_  
CI: \_\_\_\_\_  
Grupo actual de clase: \_\_\_\_\_  
Grupo actual de proyecto: \_\_\_\_\_

A través de la siguiente presento solicitud de renuncia a la nota del proyecto presentado años anteriores sabiendo que una vez presentada dicha renuncia no podré dar marcha atrás con mi decisión.

\_\_\_\_\_  
FIRMA