



**ANEP**

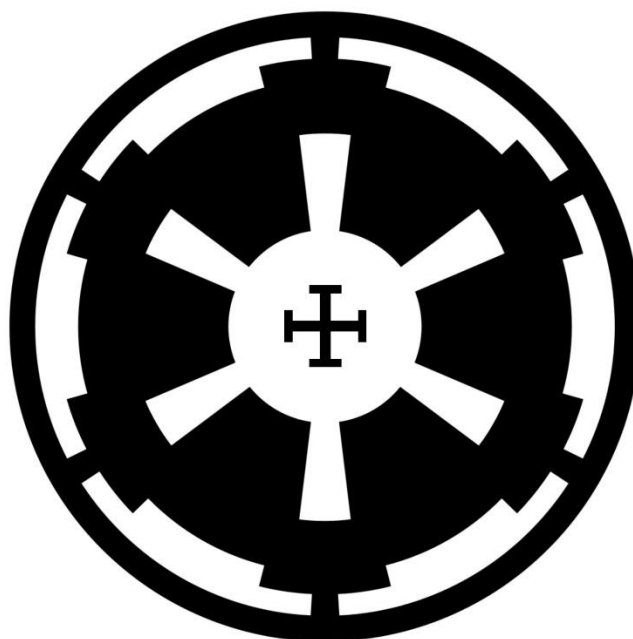


**UTU**

DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONAL



Instituto Tecnológico Superior  
**UTU**



# Primera entrega de ADA

**Solicitante:**

**I.T.S. – Instituto Tecnológico Superior Arias - Balparda**

**Nombre de Fantasía del Proyecto:**

**Imperial Software & Hardware**

**Grupo de Clase:** 3°IA

**Turno:** Matutino

**Materia:**

Análisis y Diseño de  
Aplicaciones

**Nombre de los Integrantes del Grupo:**

Esteban Saura, Santiago Couto, Mathias Ribeiro, Bruno Silva

**Fecha de entrega:** 25/7/2022

**Instituto Tecnológico Superior Arias Balparda**  
*Gral. Flores 3591 esq. Bvar. José Batlle y Ordoñez - Montevideo*



---

## Indice

|   |           |
|---|-----------|
| <b>Objetivo.....</b>  | <b>3</b>  |
| <b>Alcance .....</b>  | <b>3</b>  |
| <b>Limitaciones .....</b>   | <b>3</b>  |
| <b>Acrónimos, abreviaciones y definiciones.....</b>                             | <b>3</b>  |
| <b>1) Justificación de las técnicas de relevamiento de datos elegidas .....</b> | <b>4</b>  |
| 1.1) Entrevista.....  | 4         |
| 1.2) Cuestionario. ....   | 5         |
| <b>2) Formulario .....</b>  | <b>5</b>  |
| <b>3) Requerimientos no funcionales.....</b>                                    | <b>6</b>  |
| <b>4) Requerimientos funcionales .....</b>                                      | <b>7</b>  |
| <b>5) Roles y perfiles de usuarios .....</b>                                    | <b>9</b>  |
| 5.1) Product owner .....  | 9         |
| 5.2) Tester. ....   | 10        |
| 5.3) Developer. ....  | 11        |
| 5.4) Product manager.....   | 12        |
| 5.5) System architect. ....   | 13        |
| <b>Webgrafía.....</b>   | <b>14</b> |

**ANEP****UTU**DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONAL

---

## Propósito:

Que permita a las personas relacionadas a nuestra cooperativa (tanto productores como clientes) contar con un software para la producción agrícola. A los productores notificar próximas siembras, rotación y cosechas, la cantidad de insumos a comprar previo a la siembra de cada cultivo (tomar en cuenta los siguientes insumos: Semilla; fertilizantes) dado que, para el caso de herbicidas, fungicidas e insecticidas es necesario recorrer cada establecimiento y analizarlos de manera particular. Para los productores, además, proveer una plataforma para la compra y venta de productos.

## Alcance

La herramienta de gestión de productos agrícolas a desarrollarse proveerá la posibilidad de registrar el calendario de siembra y fertilización de los productos agrícolas, así como brindar una plataforma para la venta de semillas a productores y clientes, y como realizar consultas sobre esta información. Además, se podrá realizar la modificación de los datos.

## Limitaciones

Software:

- ☐ Visual C#.NET, MariaDB ver. 5.5.68, Workbench 6.3.4, Centos Linux 7
- ☐ Redes y Seguridad: Software de Monitoreo se puede optar entre: PANDORA FMS, ZABIX o MANAGE ENGINE - OPMANAGER

Tiempo: a definir

Costo: Omitido

## Acrónimos, abreviaciones y definiciones

RF: Requerimiento Funcional.

RNF: Requerimiento no funcional.

PM: Product manager

ICT: Tecnologías de la Información y las Comunicaciones (ICT por sus siglas en inglés, Information and Communication Technology).

OT: Tecnologías de las Operaciones (OT, Operational Technology).

**ANEP****UTU**DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONALInstituto Tecnológico Superior  
**UTU**

Look and feel: Hace referencia a la apariencia y al estilo del programa, su traducción directa del inglés sería Ver y sentir (Básicamente como se siente el usuario con respecto a la apariencia del programa).

Login: Pantalla de ingreso al programa donde el usuario ingresa sus datos (Cédula y Contraseña).

## **1) Justificación de las técnicas de relevamiento de datos elegidas**

### **1.1) Entrevista**

La entrevista nos permitirá conocer, calificar, o limitar los problemas a resolver del proyecto. También podremos obtener información sobre quiénes son los actores involucrados. Construiremos confianza con los clientes al tratar y establecer los requisitos del sistema sin ambigüedades.

En nuestro caso es mucho más útil ya que somos un grupo de trabajo pequeño, por lo que todos los integrantes podemos conocer y comprender el problema a resolver y sus implicancias. La misma es aplicable a cualquier persona ya sea analfabetos niños, o personas con limitaciones físicas o psicológicas. Nos permite estudiar aspectos de cualquier índole donde se desee profundizar en el tema; por ende, nos permite obtener información más completa y a través de ella podemos aclarar el propósito del estudio, especificar claramente la información que necesitamos y aclarar preguntas.

En cuanto a la entrevista tenemos dos tipos que podemos utilizar: la estructurada y la no estructurada.

En la entrevista estructurada realizamos un formulario estandarizado en el cual se definen preguntas, las cuales son las mismas y mantienen el mismo orden para todos los entrevistados, en las preguntas se dan opciones de 2, 3 o pocas más alternativas y los comentarios serán los mismos para todos.

La entrevista estructurada nos otorga respuestas cortas y precisas, por ende, la información se nos hace mucho más fácil de procesar y la información es mucho más uniforme.

En la entrevista no estructurada poseemos una más flexibilidad, pero es regida por los objetivos de la investigación, las preguntas de la misma, su orden y formulación son controladas por el investigador, el cual puede adaptarlas dependiendo de la situación y características del entrevistado. El mismo cuenta con libertad para dar sus respuestas, lo que permite obtener más información, es adaptable y aplicable a toda clase de sujetos en diversas situaciones y podremos profundizar en temas de interés.

**ANEP****UTU**DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONALInstituto Tecnológico Superior  
**UTU**

---

## 1.2) Cuestionario

El cuestionario nos resulta útil de forma masiva, podemos aplicarlo a muchas personas por un bajo costo. En el caso de que utilicemos cuestionarios online, no necesitaremos gastar en papel ni en espacio, para guardar las respuestas.

Nos permite obtener resultados fáciles de cuantificar. Podremos realizar gráficos que nos ilustren qué respuestas fueron más comunes y tomar decisiones a raíz de ello.

A la hora de realizar el cuestionario tendremos diferentes tipos de preguntas las cuales son: abiertas, cerradas y mixtas.

En las preguntas abiertas las personas contestan con sus propias palabras, permitiendo total libertad en la respuesta, expresan sus ideas sin limitaciones. Un ejemplo sería:

¿Cómo te gustaría que fuera la interfaz del programa?

Escriba su respuesta:

---

Luego podremos utilizar las preguntas cerradas las cuales son un conjunto de opciones predefinidas, las cuales son muy precisas, rápidas de contestar y son muy fáciles de analizar al visualizar los resultados. Y por último las preguntas mixtas las cuales son una combinación de las preguntas abiertas y cerradas. Básicamente son preguntas con opciones de respuesta cerradas, a las cuales se le agrega una opción abierta al final, de forma que la persona pueda escribir su respuesta si no encuentra ninguna apropiada entre las predefinidas, o que pueda explicar su anterior respuesta corta.

## 2) Formulario

¿Cuántos switch desea en total?

- 1
- 2
- 3

¿Qué rack prefiere?

- Rack aéreo
- Rack terrestre

¿Cuál de las siguientes marcas prefiere?

- Cisco
- TP-Link
- ZTE

---

¿Cuál de estos dos temas prefiere para la visualización del programa?

- Tema claro (Fondo blanco con letra negra)
- Tema oscuro (Fondo negro con letra blanca)

¿Prefiere un fondo solido o una imagen relacionada con la temática de la cooperativa?

- Fondo solido
- Imagen

¿En cuál de las siguientes posiciones le gustaría que se ubique el botón de salir?

- Arriba a la derecha
- Abajo a la derecha

¿Qué sistema operativo prefiere para el servidor?

- Linux
- Windows

### 3) REQUERIMIENTOS NO FUNCIONALES

**Look and feel:**

El logo debe ser minimalista.

**Usabilidad y Humanidad:**

Interfaz sin excesivas decoraciones para que el usuario no se confunda.

**Rendimiento:**

Avisar si se encuentra un error al estar ejecutando el programa.

**Operación:**

Manual de usuario del programa.

**Mantenimiento y soporte:**

El sistema estará disponible para Windows.

**Seguridad:**

Dar la opción de mostrar o no la contraseña al ingresarla.

**Culturales y políticos:**

No incluir referencias a símbolos partidarios.

**Legales:**

El producto debe contar con licencia de software vigente.

## 4) Requerimientos funcionales

| Numero | Nombre   | Descripción   |
|--------|----------|---|
| RF1    | Login    | El sistema deberá contar con una instancia donde se pueda ingresar cédula y contraseña. Según el tipo de usuario se abrirán ventanas específicas. Se encontrarán dos variables. Una de tipo String una para guardar la contraseña ingresada y una de tipo entero para la cédula.            |
| RF2    | Registro | El sistema deberá permitir que se ingresen datos y si el identificador no está registrado, que se añadan a la base de datos. Se encontrarán dos variables. Una de tipo String una para guardar la contraseña ingresada y una de tipo entero para la cédula.                                 |
| RF3    | Compra   | Debe contener una sección que permita realizar transacciones en donde el productor entregue una determinada cantidad de dinero a cambio de una determinada cantidad de producto. Tendremos una variable de tipo Int para que el usuario ingrese la cantidad de artículos que quiere comprar |



|      |                            |  |
|------|----------------------------|--|
| RF4  | Venta                      | Debe contener una sección que permita realizar transacciones en donde el productor entregue una determinada cantidad de producto a cambio de una determinada cantidad de dinero. |
| RF5  | Notificación de cosecha    | A través del usuario administrador se fijará el calendario de siembra y se notificará mediante un aviso a los productores.   |
| RF6  | Gestión de usuarios.       | El administrador podrá agregar, eliminar y modificar productores mediante el ingreso de determinados datos, principalmente la cédula.  |
| RF7  | Gestión propia             | Cada productor podrá modificar sus propios datos.  |
| RF8  | Criterios de fertilización | Los productores podrán establecer criterios de fertilización con nitrógeno.  |
| RF9  | Gestión de productos       | Los productores podrán gestionar los datos de sus productos y predios.   |
| RF10 | Historial de transacciones | Habrà un apartado donde se encuentre una lista con las compras y ventas realizadas por el productor, junto a los datos de esas transacciones.                                    |



---

## 5) Roles y perfiles de usuarios

### 5.1) Product owner

El Product Owner o propietario del producto se enfoca en entregar el mejor producto posible a los usuarios finales. Para hacerlo, los propietarios desarrollan una visión de cómo debería funcionar el producto, definen características específicas y transforman esas características en tareas del trabajo pendiente (backlog) del producto, para que el equipo de Scrum trabaje en ellas. El Product Owner es responsable del producto final y actúa como nexo entre las partes interesadas del negocio, los miembros del equipo y los usuarios finales.

Materias que aplica:

ADA: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

Gestión de proyectos: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

Tareas:

#### **Co-creación de la visión del producto**

El Product Owner es responsable de co-crear una visión que quede en claro: quiénes son nuestros clientes, qué problema les vamos a resolver y beneficios clave del producto.

#### **Gestión del Product Backlog**

El Product Owner es el responsable de que el Product Backlog sea visible y transparente, debe ser visible para stakeholders y equipo, debe estar expresado claramente en sus ítems. Product Backlog Items deben estar especificados de manera clara y estar lo suficientemente detallados para ser entendidos por el equipo y por último tiene que estar ordenado de manera que maximice el valor del producto creado por el equipo.

#### **Ordena el Product Backlog**

Se encarga de ordenar los ítems del Product Backlog de manera de maximizar la entrega de valor del equipo.

#### **El Product Owner maneja las expectativas de los stakeholders**

El Product Owner pasa gran parte de su tiempo con los stakeholders (interesados), teniendo conversaciones sobre el producto, su visión, características y funcionalidades. La comunicación efectiva es una competencia clave ya que tiene que poder comunicarse efectivamente con múltiples stakeholders y el equipo.

#### **Release Plan y estrategia de producto**

El Product Owner tiene un plan de entregas en el cual refleja la visión del producto a lo largo del tiempo. Un Release Plan contiene entre otras cosas:

- La visión del producto descompuesta en objetivos, funcionalidades y épicas.
- Un Release Burndown Chart que muestre de qué manera avanzamos hacia la entrega del release.
- Un Roadmap que muestre la cronología de entrega de funcionalidades y épicas a futuro, tiene el objetivo de facilitar conversaciones con stakeholders.
- El Release Plan no es fijo, sino que cambia de manera continua y se adapta a los cambios de contexto.



---

## 5.2) Tester

Los Tester planifican y llevan a cabo pruebas de software de los ordenadores para comprobar si funcionan correctamente. Identifican el riesgo de sufrir errores de un software, detectan errores y los comunican. Evalúan el funcionamiento general del software y sugieren formas de mejorarlo.

Los probadores de software pueden probar todo tipo de software, programas individuales para aplicaciones o productos (una serie de programas que almacenan y procesan información para realizar una tarea específica). Las aplicaciones incluyen pantallas para que los usuarios introduzcan información y puedan imprimir los resultados. Los probadores también pueden analizar sitios web o juegos de ordenador.

Pueden estar involucrados con el desarrollo de un nuevo programa, ya que a menudo deben probar nuevas versiones de un software, o simplemente llevar a cabo un control rutinario de un producto existente.

En primer lugar, identifican todos los riesgos tales como: errores que podrían ocurrir en el software cuando alguien lo está usando, confusiones a la hora de utilizar un software por parte de un usuario debido a la falta de información, la instalación del software en diferentes tipos de sistemas, etc. Prueban estos riesgos para conseguir que un software ofrezca la máxima confianza. Deben diseñar pruebas que se puedan repetir y evaluar.

Pueden evaluar el correcto funcionamiento:

- Si el software coincide con la especificación original (o qué hay que hacer para que coincida).
- Si cuando un usuario hace clic en un botón o enlace para que suceda algo, por ejemplo, para ir a una nueva pantalla, se abre la pantalla correcta.
- Si el usuario tiene que introducir algunos datos, el software actúa de forma correcta y los datos se almacenan de forma segura.
- Si está diseñado para que varios usuarios puedan abrirlo al mismo tiempo sin que el programa reduzca su velocidad de procesamiento.
- Si se puede utilizar herramientas de automatización de pruebas para iniciar una sesión en muchos usuarios al mismo tiempo.
- Si cuando el software tiene que compartir información con otro programa, continúa funcionando correctamente.

---

Los probadores de software, por ejemplo, pueden introducir deliberadamente información en una ruta equivocada para asegurarse de que reciben el mensaje de error esperado. Por ejemplo, si se les pide que introduzcan un año de nacimiento, podrían poner 1892 en lugar de 1992 y comprobar que el programa reacciona de la forma adecuada y detecta el error. No suelen tener tiempo para probar todas las combinaciones posibles de las acciones que pueden aplicarse con un software, por lo que tienen que ponerse de acuerdo de antemano con el cliente sobre el tiempo disponible que tendrán. Mantienen un registro detallado las pruebas que realizan. Una vez completadas las pruebas, enumeran los errores y redactan un informe para que los programadores y administradores de proyectos puedan implementar posteriormente.

Algunos probadores trabajan como consultores, y deben viajar a los hogares/locales de los clientes para asesorarles sobre los procedimientos de prueba, así como la realización de las pruebas.

Materias que aplica:

Programación: Esteban Saura, Santiago Couto

ADA: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

Gestión de proyecto: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

## 5.3) Developer

El Developer es el profesional responsable de crear los entregables del proyecto, junto con el resto del equipo.

El mismo se encarga de la comprensión de los requisitos comerciales especificados por el propietario del producto. Estimar las historias de usuario en la acumulación de sprint, desarrollo del producto/servicio (entregables).

Cada Developer es responsable de entregar un elemento funcional (también conocido como fragmento) del producto al final de cada sprint.

Tiene que asistir a las reuniones informando las tareas planificadas para cada día.

Asegurar al propietario del producto y al Scrum Master que el trabajo asignado se está realizando según lo planeado, Garantizar una comprensión clara de los requerimientos del sistema, proporcionar información sobre la creación de historias de usuario al propietario del producto, Informar al Scrum Master de los impedimentos para el progreso del proyecto. Proporcionar información sobre la creación de historias de usuario al propietario del producto, comprender las historias de usuario y estimar historias de usuarios aprobadas por el propietario del producto

Materias que aplica:

Gestión de proyecto: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

ADA: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

---

## 5.4) Product manager

El Product manager es la figura que tiene la responsabilidad de identificar las necesidades de los consumidores y satisfacerlas a través del desarrollo y entrega de productos. Esto incluye la parte de investigación previa al desarrollo del producto, las estrategias y el plan de marketing posteriores al mismo para que la recepción por parte del mercado sea lo más óptima posible.

El PM es un puesto que se encuentra entre los departamentos de negocio, tecnología y experiencia de usuario, trabajando como intermediario y puente entre dichos equipos.

PM debe tener un buen contacto con el departamento de desarrollo. Esto es porque el PM ha de ser capaz de entender qué problema solucionar y el por qué ha de hacerse. Será su función coordinarse con los equipos de desarrollo de producto y programadores para entender cómo ha de crearse dicho producto.

El PM también se encarga de:

- Identificar qué problemas y necesidades tiene el consumidor para saber qué productos crear.
- Garantizar que los productos se creen y entreguen correctamente teniendo en cuenta la propuesta de valor.
- Analizar cómo responden los usuarios a dichos productos y obtener insights para mejorarlos constantemente.
- Optimizar los procesos anteriores e identificar constantemente puntos de mejora en los diferentes procesos.
- Definir una visión para el producto y un claro roadmap con entregables en el tiempo para testar el valor con el mercado.
- Analizar periódicamente las necesidades de los usuarios para mejorar la entrega de valor. Esto a través de metodologías tanto cualitativas como cuantitativas.
- Analizar las macrotendencias de su sector y tendencias en la tipología de producto para tener una solución actualizada.
- Coordinar y monitorizar la percepción de la marca y del producto por los diferentes segmentos de clientes.
- Priorizar las funcionalidades del producto traduciéndolo a ROI. Es decir, se priorizarán aquellas funcionalidades que generen un mayor retorno en las cuentas de la empresa.

Materias que aplica:

Formación empresarial: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

Gestión de proyecto: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

ADA: Esteban Saura, Mathias Ribeiro, Bruno Silva, Santiago Couto

---

## 5.5) System architect

Los arquitectos de sistemas diseñan la arquitectura de los sistemas informáticos para que estos cumplan los objetivos y necesidades que motivan su desarrollo.

La función principal de un arquitecto de sistemas es el diseño de la arquitectura de los sistemas tanto de Tecnologías de la Información y las como de Tecnologías de las Operaciones.

El propósito de los sistemas ICT y OT suele ser amplio y, por lo tanto, complejo de modelar y construir como un todo. Por eso, el diseño de su arquitectura consiste en:

Descomponer la funcionalidad global del sistema en distintos componentes/subsistemas más manejables, los cuales se encargarán de una pequeña parte de la funcionalidad.

Definir las interacciones e interfaces entre estos (integración).

Decidir las tecnologías y recursos necesarios para su construcción.

Los arquitectos de sistemas (denominados también arquitectos de software) diseñan y lideran la construcción de sistemas informáticos, los cuales operan como parte de la infraestructura de una organización y permiten que funcione. Así, estos profesionales definen cómo se deben usar los sistemas, cuánto estrés pueden soportar y qué es necesario para que estos operen en distintas situaciones y condiciones.

Para el diseño de los sistemas, consideran los objetivos de negocio a los que deben dar respuesta, determinan su complejidad y dimensionamiento (recursos informáticos y presupuesto), la experiencia del usuario final que los va a utilizar y los requisitos para asegurar la disponibilidad, confidencialidad e integridad de la información que traten.

Un arquitecto de sistemas diseñará toda la infraestructura técnica necesaria para el funcionamiento del sistema, hardware, software y necesidades de comunicación, e integrará los requisitos de seguridad de la información en el diseño del sistema.

Un arquitecto de software también se encarga de:

- Diseño del sistema a nivel funcional y técnico con el objetivo de que satisfaga las necesidades de negocio, de experiencia de usuario y de calidad (requisitos del sistema).
- Integrar desde el diseño del sistema los requisitos de seguridad de la información.
- Documentar los pasos necesarios para su creación alineando buenas prácticas y estándares de referencia.
- Guiar al equipo de desarrollo e integración.
- Supervisar la construcción del sistema, pruebas de funcionamiento y documentación.
- Verificar la conformidad del sistema en cuanto a los requisitos de calidad, funcionales de los interesados y de seguridad de la información

Materias que aplica:

Programación: Santiago Couto, Esteban Saura.

Sistemas Operativos: Bruno Silva, Mathias Ribeiro.



ANEP



UTU

DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONAL



Instituto Tecnológico Superior  
UTU

---

## Webgrafía

Developer:

<https://www.qrpinternational.be/blog/glossary/who-is-the-scrum-developer-role-and-responsabilities/>

Product Manager:

<https://www.iebschool.com/blog/que-hay-de-los-product-manager-comercio-ventas/>

Product Owner:

<https://ittude.com.ar/b/scrum/product-owner/>

System Architect:

<https://www.unir.net/ingenieria/revista/arquitecto-sistemas/>

Tester:

<https://www.hiberus.com/crecemos-contigo/testing-fase-de-testeo-de-software/#:~:text=Qu%C3%A9%20es%20un%20tester,detectan%20errores%20y%20los%20comunican>

