**Especificación de Requisitos según el estándar**

**de IEEE 830**

**IEEE Std. 830-1998**

**29 de Septiembre 2022**

**Resumen.**

Este documento presenta, en castellano, la definición de los requerimientos del sistema basado en historias de usuario de la aplicación web de Administración de turnos para emprendedores.

Dicha aplicación brinda al emprendedor la capacidad de informar a sus clientes la disponibilidad de sus turnos de su emprendimiento y obtener una agenda organizada de manera automática, mientras por el lado del cliente, da la posibilidad de acceder a esos turnos y generar reservas.

**INDICE**

1. **Introducción** 3
   1. Propósito 3
   2. Ámbito del sistema 3
   3. Definiciones, Acrónimos y Abreviaturas 3
   4. Referencias 3
   5. Visión general del documento 3
2. **Descripción general** 4
   1. Perspectiva del producto 4
   2. Funciones del producto 4
   3. Característica de los usuarios 4
   4. Restricciones 4
   5. Suposiciones y dependencia 5
   6. Requisitos futuros 5
3. **Requisitos específicos** 5
   1. Requerimientos funcionales 5
   2. Requerimientos no funcionales 8
   3. Requisitos de rendimiento 8
   4. Restricciones de diseño 8
   5. Atributos del sistema 9
4. **Anexo** 9
   1. Presentación de sprints 9
   2. Historias de usuarios 11
   3. Conclusión 15
5. **Introducción.**
   1. **Propósito.**

Por medio de este documento especificaremos los requerimientos que cumplirá el E-commerce entre emprendedores y clientes, y además en un futuro se implementará un sistema de Administración de turnos para emprendedores.

Está dirigido a desarrolladores que se ocupen del análisis, diseño e implementación, así también a quienes desarrollen el testing del software.

* 1. **Ámbito del sistema.**

El sistema será definido como TurNow Ecommerce.

Proporcionará un comercio electrónico en el cual los emprendedores van a poder publicar sus productos especificando todos los detalles, y los clientes lo pueden adquirir mediante una forma de pago electrónica. Además posteriormente se integrará una gestión de turnos para el emprendedor, generándole un reporte sobre sus turnos diarios. Por el lado del cliente, le brindará información sobre cada emprendedor (trabajos que realizan, ubicación, etc.) y a su vez dispondrá de la posibilidad de generar una reserva para ser atendido.

Los beneficios que se obtendrán al usar este sistema serán:

* Compras y ventas de productos.
* Compras online de insumos.
* Publicar turnos disponibles y dar posibilidad de reservarlos por el cliente.
* Brindar una difusión de los emprendimientos y acceso a la información sobre ellos a los clientes.
* Ordenar la agenda del emprendedor de manera automática.

El sistema será desarrollado como aplicación web para cumplir con el objetivo de gestionar los turnos de emprendimientos de manera automática.

* 1. **Definiciones, Acrónimos y Abreviaturas.**
* ERS. Especificación de requerimientos de software.
* RF. Requerimiento funcional.
* RNF. Requerimiento no funcional.
  1. **Referencias.**

Este documento se ha desarrollado basándose en el Standard IEEE 830. Este estándar es una guía para definir los requerimientos que cumplirá el sistema.

* 1. **Visión General del Documento.**

En este documento se da a conocer cuál será el funcionamiento del sistema. Se establecerá una explicación gráfica y escrita de cada una de las funcionalidades que el sistema debe cumplir. Se mostrará si es que el sistema tiene limitaciones, y se explicará el porqué de estas. Definiremos quienes van a interactuar directamente con el sistema.

1. **Descripción general.**
   1. **Perspectiva del producto.**

El sistema se enfocará en el E-commerce y no tendrá interacción o dependencia con otros programas que estén operando, exceptuando el navegador web a utilizar.

* 1. **Funciones del producto.**

El sistema facilitará la gestión de:

* Registro de Emprendedores/emprendimiento.
* Registro de clientes.
* Publicación de productos.
* Compra y venta de productos.
* Proceso de turnos.
* Generación de cartilla de emprendimientos.
  1. **Características de los usuarios.**

El sistema es manejado por emprendedores y clientes.

Emprendedor:

Nivel de educación: Media.

Experiencia técnica: Media.

Cliente:

Nivel de educación: Media.

Experiencia técnica: Media.

El sistema va a estar enfocado en la facilidad de uso, permitiendo así que los usuarios no necesiten un nivel alto de conocimientos, de igual manera, se generara la documentación y demostración para su uso correcto.

* 1. **Restricciones.**

Interfaz para ser usada con internet.

Las compras se harán de manera virtual a través de la tienda destinada a tal fin.

Se requiere Uso de un Dominio.

El servidor debe ser capaz de atender consultas concurrentemente.

El sistema deberá tener un diseño e implementación sencilla, independiente de la

plataforma o del lenguaje de programación.

* 1. **Suposiciones y dependencias.**

El sistema se desarrollará en navegador web, ciertos requerimientos se verán afectado a la compatibilidad de estos y a futuras actualizaciones.

* 1. **Requisitos futuros.**

Como requisitos futuros:

* Compatibilidad con Mobile.
* Agregar al perfil de cliente una billetera de puntos. Al consumir alguno de los emprendimientos se acumulan para luego ser cambiados por algún beneficio.
* Sistema de calificación por estrellas de emprendimientos realizadas por clientes, posible inclusión de comentarios.

1. **Requisitos específicos.**
   1. **Requerimientos funcionales.**

**RF1. Administrar emprendimiento/emprendedor.**

Para cumplir este requerimiento, se debe cumplir los siguientes aspectos:

* Crear emprendedor: Mediante un formulario se debe solicitar la información para generar un perfil. Se adhieren los datos del emprendimiento (Nombre, descripción de servicio/producto, turnos, etc.).
* Consultar emprendedor: Por medio de esta función se podrá acceder a los emprendedores que están dados de alta y a su información correspondiente.
* Modificar emprendedor: Esta función permite la edición de los datos cargados previamente.
* Eliminar emprendedor: a través de esta función se elimina el registro de un emprendedor.
* Crear artículos para la venta
* Posibilidad de comprar artículos que estén en venta de otros emprendedores/emprendimientos.

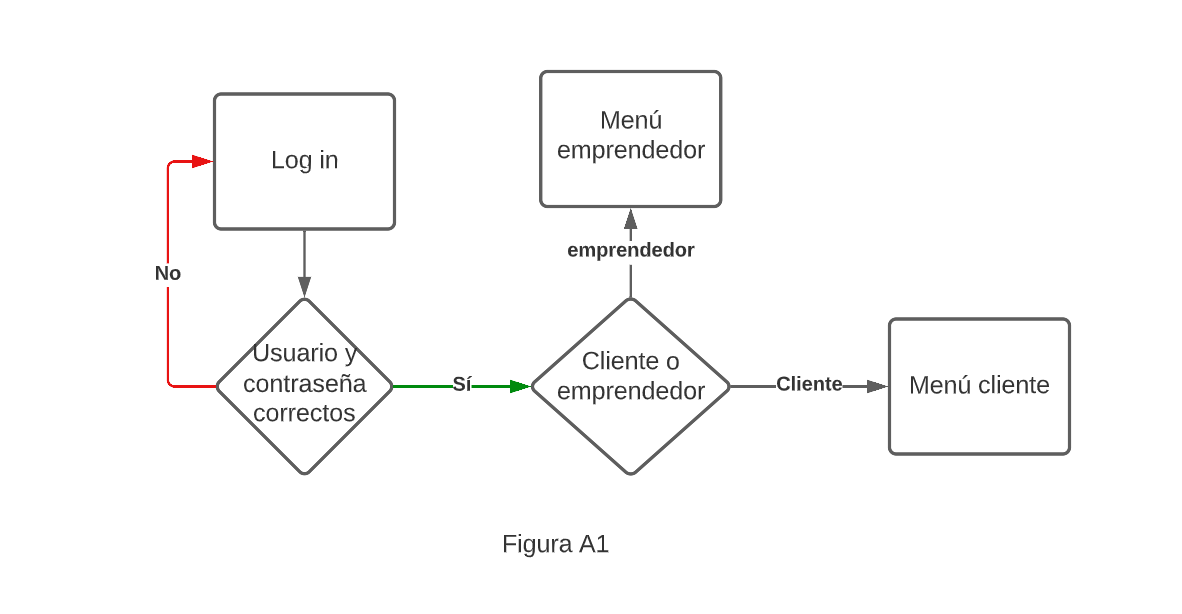
**RF2. Administrar clientes.**

Para cumplir este requerimiento, se debe cumplir los siguientes aspectos:

* Crear cliente: Mediante un formulario se debe solicitar la información para generar un perfil.
* Consultar cliente: Por medio de esta función se podrá acceder a los clientes que están dados de alta y a su información correspondiente.
* Modificar cliente: Esta función permite la edición de los datos cargados previamente.
* Eliminar cliente: a través de esta función se elimina el registro de un cliente.
* Comprar artículo en venta de emprendedores/emprendimientos.

**RF3. Log in/Log out.**

Para cumplir este requerimiento de log in, la aplicación debe tener una página para ingresar usuario y contraseña. Si este usuario existe, lo clasifica como cliente o emprendedor, redireccionandolo a la página que corresponda (Figura A1).



En el caso del log out, al ya estar el usuario con la sesión iniciada, la página debería mostrar un botón para poder desloguearse.

**RF4. Crear producto para venta (E-COMMERCE)**

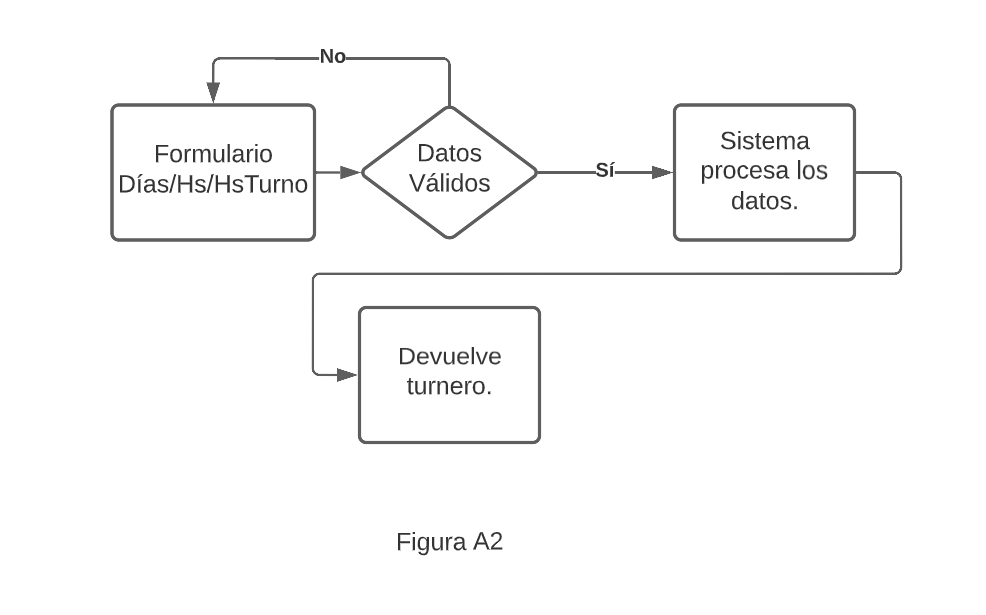
Para cumplir este requerimiento, el emprendedor debe acceder a su cuenta y crear desde el menú vender un nuevo producto para publicar en la plataforma de venta. Dicho producto deberá tener: nombre, descripción, imagen, precio, cantidades a la venta y se asociara a un emprendedor y a una categoría.

**RF5. Comprar producto en venta (E-COMMERCE)**

Tanto el cliente como el emprendedor podrá acceder a la plataforma de venta para realizar compras de productos publicador por los emprendedores/emprendimientos.

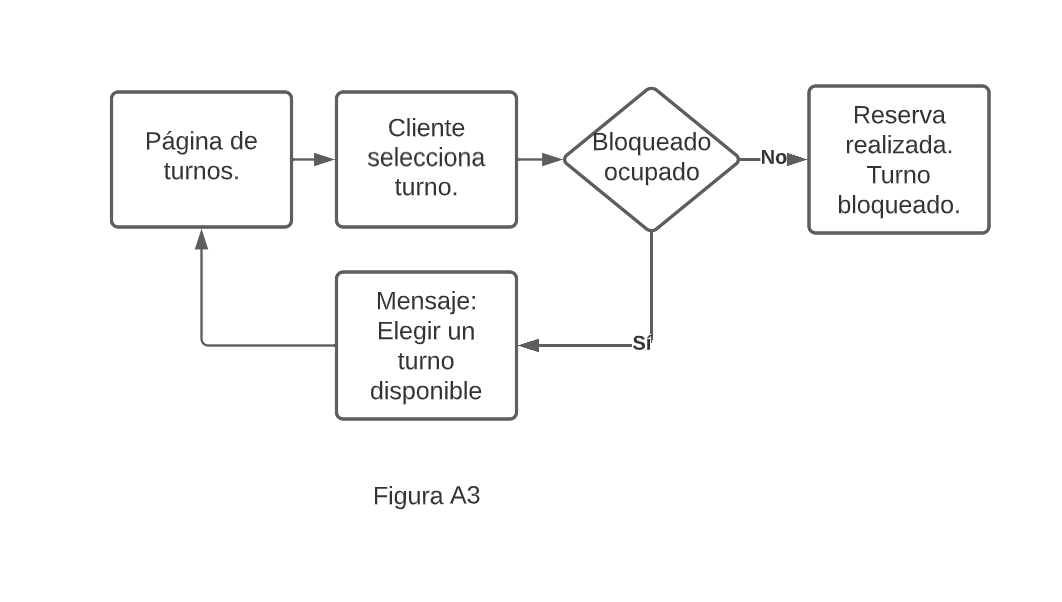
**RF6. Administrar disponibilidad de turnos.**

Para cumplir este requerimiento, al ingresar un emprendedor se le solicitará mediante formulario el ingreso de días dispuestos a trabajar y horarios, conjuntamente con unidad de tiempo mínima al usarse en turno. Al tener acceso a esta información, el sistema puede calcular y mostrar los turnos disponibles del emprendedor (Figura A2).



**RF7. Administrar turnos disponibles.**

Para cumplir este requerimiento, al ingresar un cliente a la página de turnos de algún emprendedor, se le debe mostrar los días y horarios disponibles. Estos deben ser seleccionables y luego de ser seleccionados, bloqueados para no tener una futura reserva en el mismo horario (Figura A3).



**RF8. Mostrar turnos a emprendedor.**

Para cumplir este requerimiento, el emprendedor debe acceder a una visualización de sus turnos actualizada desde una página.

* 1. **Requerimientos no funcionales.**

**RNF1. Lenguaje de programación.**

La aplicación será desarrollada en lenguaje de etiquetas HTML, CSS, lenguaje de programación JAVASCRIPT, ANGULAR (framework de front-end), y Django (framework para back-end).

**RNF2. Base de datos.**

La aplicación será desarrollada con bases de datos relacional MySql.

**RNF3. Restricciones de funcionamiento.**

Para el acceso al sistema existen dos tipos de usuarios:

* Emprendedor: tendrá acceso a funcionalidades y características exclusivas, por ejemplo: detallar sus datos, publicar productos para la venta, configuración de turnos, etc. No podrá editar datos de cliente.
* Cliente: tendrá acceso a su configuración de perfil, y ver información de los emprendimientos y artículos en venta en la tienda online para su compra; y acceso a solicitar una reserva de turnos. No podrá modificar datos de emprendedor.
  1. **Requisitos de rendimiento.**

Al ser un proyecto en desarrollo, es muy difícil estimar cual van a ser los requisitos de rendimiento. De manera estimativa y buscando una posible realidad, suponemos que:

* Usuarios conectados simultáneamente: < 50.
* Transacciones por segundo: < 1.
* Cantidad de productos < 3000.
* Cantidad de usuarios registrados: <1000
* Cantidad de turnos registrados (Turnos): < 1000000.

Los requisitos de rendimiento se verán afectados por la difusión de la aplicación, posiblemente estos números estimados sean irreales en un futuro, pero son acertados como iniciales.

* 1. **Restricciones de diseño.**

Tener en cuenta que en el desarrollo del diseño de nuestro software no se debe anteponer con el resto de los requerimientos y restricciones citadas en puntos anteriores para que no genere ningún conflicto en su ejecución.

Otro punto importante a tener en cuenta es que los requerimientos de hardware hacia el emprendedor/cliente sean mínimos a fin de generar una buena experiencia de usuario.

* 1. **Atributos del sistema.**

El sistema realizará sus funciones de manera correcta, determinado por un mantenimiento constante por el grupo de desarrolladores. Testeando las funcionalidades previamente, antes de llegar a producción.

Referido al tema de portabilidad, la aplicación puede ser ejecutada por cualquier navegador web actual.

En seguridad, se establecerá un sistema de login/logout para verificar usuarios. A través de Email y/o teléfono móvil.

1. **Anexo**
   1. **Presentación de sprints.**

**SPRINT 0.**

Duración del sprint: (10/04 – 21/04)

* Agregar Issues como Historia de Usuarios, respetando la nomenclatura sugerida.
* Crear Milestone por Sprints.
* Crear Wiki con registro de ceremonias por SPRINT (daily, review,retrospective) y novedades del equipo.
* Completar e incluir la documentación IEEE830 con la nueva parte e-commerce.
* Crear tablero kanban (Product Backlog, ToDo, In process, Finished).
* Crear diagramas casos de uso.
* Crear diagramas de clase.
* Mejorar el modelo de datos lógico en base a los nuevos requerimientos (e-commerce).
* Revisar la estructura web semántica, responsive (RWD).

**SPRINT 1.**

Duración del sprint: (24/04 – 07/05)

* Mantener actualizado el flujo de las Issues.
* Crear SPA con módulos, componentes y routin en Angular.
* Implementar las clases en Python en base al diagrama de clases.
* Diseñar un script legible de la base de datos en MySQL
* Corregir, si es necesario, el diagrama casos de uso
* Corregir, si es necesario, el diagrama de clases
* Corregir, si es necesario, el diagrama de entidad-relación
* Corregir, si es necesario, el modelo relacional.
* Realizar las tablas actualizadas de la normalización.

**SPRINT 2.**

Duración del sprint: (08/05 – 21/05)

* Crear servicios que consuman un API Rest de prueba.
* Inyectar a los componentes los correspondientes servicios a fin de que puedas acceder a los datos y mostrarlos por pantalla.
* Incorporar la programación reactiva mediante observables para la comunicación entre servicios y componentes.
* Establecer el modelo de base de datos del primer sprint en Django usando clases.
* Establecer el sistema de usuarios en Django-Admin (registro, login y roles).
* Crear un ABM de productos/servicios utilizando el panel de control de Django.

**SPRINT 3.**

Duración del sprint: (22/05 – 04/06)

* Incorporar formularios reactivos de registro y login para la autenticación.
* Incorporar formularios para la manipulación (alta) de servicios/productos/suscripción.
* Conexión del front end con el back end para el registro e inicio de sesión.
* Conexión del front end con el back end para almacenar los datos de los productos/servicios y detalles de producto/servicio en la DB a través del modelo en Django.
* Crear registro y login de usuario desde el back end.
* Alta de al menos 3 productos/servicios/suscripción.
* Crear un dashboard con módulos planteados en la maquetación del front end.

**SPRINT 4.**

Duración del sprint: (05/06 – 18/06)

* Crear un tablero de control para usuarios.
  + Mostrar catálogo con mínimo 3 productos
  + Seleccionar para comprar
  + Simular pagon con API Mercado Pago
* Crear un tablero de control administrador CRUD básico funcional de producto y cliente.
* Demo final.
  1. **Historias de usuarios.**

Para la organización de cada actividad de los sprints, utilizamos las "issues" que nos proporciona GitHub. Hemos destacado 2 códigos para diferenciarlas: "#US" para aquellas issues destinadas a ser Historias de Usuarios y "#TK" para aquellas issues destinadas a ser Tareas. Las Historias de Usuarios representan las actividades generales a realizar, mientras que las Tareas son elementos adicionales necesarios para completar dichas actividades.

Además, cada Historia de Usuario estaba calificada con unos "labels". Estos labels se dividen en 3 grupos:

* Prioridad:
  + Prioridad Alta
  + Prioridad Media
  + Prioridad Baja
* Tipo de issue:
  + Historia de Usuario
  + Tarea
* Área de trabajo:
  + Documentación
  + Frontend
  + Backend

De esta manera, utilizamos las issues y los labels en GitHub para tener una mejor organización y seguimiento de nuestras actividades en cada sprint.

La Scrum Máster se encargó de crear las issues para cada actividad en cada sprint. Al comienzo de cada sprint, se llevaba a cabo una reunión en la que se realizaba una retrospectiva del sprint anterior y una presentación del sprint actual, donde se analizaban los ejercicios a realizar. A lo largo de los días, los integrantes del equipo se reunían a través de diversas plataformas para organizarse y abordar las actividades autoasignadas.

A continuación, presentamos todas las Historias de Usuario (US) que hemos creado en cada sprint:

**Sprint 0**

* #US01 - Como integrante del equipo de trabajo Quiero disponer de un proyecto en el cual se disponga de un tablero para administrar las actividades del equipo
  + #TK01 - Generar repositorio y gestionar accesos
  + #TK02 - Crear tablero y gestionar accesos
  + #TK03 - Generar primeras Task y HU en modo ejemplo.
* #US02 - Como integrante del equipo de trabajo Quiero disponer de una ficha con información Para presentar el equipo a los profesores del ISPC
* #US03 - Como integrante del equipo de trabajo Quiero agregar la parte e-commerce al documento IEEE830 Para completar la documentación del proyecto
* #US04 - Revisar la estructura web Semántica, responsive (RWD)
* #US05 - Como integrante del equipo de trabajo Quiero revisar el diagrama de clases Para continuar con el avance del proyecto integrador
* #US06 - Como integrante del equipo de trabajo Quiero revisar el diagrama de caso de uso Para continuar con el avance del proyecto integrador
* #US07 - Como integrante del equipo de trabajo Quiero revisar el modelo relacional y actualizarlo si es necesario Para continuar con el avance del proyecto integrador
* #US08 - Como integrante del equipo de trabajo Quiero revisar el modelo ER Para continuar con el avance del proyecto integrador
* #US09 - Como integrante del equipo de trabajo Quiero revisar el mapa de sitio y actualizarlo si es necesario Para continuar con el avance del proyecto integrado

**Sprint 1**

* #US10 - Como integrante del equipo de trabajo Quiero crear un script legible de la BD en MySQL Para continuar con el avance del proyecto integrador
* #US11 - Como integrante del equipo de trabajo Quiero implementar el diagrama de clases en Python Para continuar con el avance del proyecto integrador.
* #US12 - Como integrante del equipo de trabajo Quiero diseñar la Normalización usando como referencia los diagramas realizados Para administrar continuar con el avance del proyecto.
* #US13 - Como integrante del equipo de trabajo Quiero convertir los archivos .html en una SPA en Angular con módulos y componentes Para continuar con el avance del proyecto.
* #US14 - Como integrante del equipo de trabajo Quiero crear un Draft del código fuente en GitHub Para continuar con el avance del proyecto integrador.

**Sprint 2**

* #US15 - Como integrante del equipo de trabajo Quiero aplicar servicios a los componentes Para continuar con el avance del proyecto integrador.
  + #TK04 - Incluir enlace de datos (Data binding)
* #US16 - Como integrante del equipo de trabajo Quiero crear MVC en Django Para continuar con el avance del proyecto integrador.
* #US17 - Como integrante del equipo de trabajo Quiero crear una simulación de consumir servicios mediante JSON Para continuar con el avance del proyecto integrador.
* #US18 - Como integrante del equipo de trabajo Quiero implementar formularios reactivos Para continuar con el avance del proyecto integrador.
* #US19 - Como integrante del equipo de trabajo Quiero servicios en Angular que hagan peticiones HTTP Para continuar con el avance del proyecto integrador.
  + #TK05 - Crear servicios que consuman un API Rest de prueba.
* #US20 - Como integrante del equipo de trabajo Quiero actualizar el documento IEEE830 Para continuar con el avance del proyecto integrador.
* #US22 - Como integrante del equipo de trabajo Quiero incorporar programación reactiva mediante observables Para la comunicación entre servicios y componentes.
* #US23 - Como integrante del equipo de trabajo Quiero el modelo de bbdd en Django usando clases Para continuar con el avance del proyecto integrador.
* #US24 - Como integrante del equipo de trabajo Quiero crear un ABM prod/serv utilizando panel de control Django Para continuar con el avance del proyecto integrador.

**Sprint 3**

* #US21 - Como integrante del equipo de trabajo Quiero crear una API Rest Para continuar con el avance del proyecto integrador.
* #US25 - Como integrante del equipo de trabajo Quiero diseñar formularios Para continuar con el avance del proyecto integrador.
* #US26 - Como integrante del equipo de trabajo Quiero conectar front end con bak end Para el login.
* #US27 - Como integrante del equipo de trabajo Quiero conectar front end con bak end Para almacenar datos producto/servicio.
* #US28 - Como integrante del equipo de trabajo Quiero un registro de usuario Para continuar con el avance del proyecto integrador.
* #US29 - Como integrante del equipo de trabajo Quiero crear un login de usuario Para continuar con el avance del proyecto integrador.
* #US30 - Como integrante del equipo de trabajo Quiero crear registro/alta de 3 productos/servicios/suscripciones Para continuar con el avance del proyecto integrador.
* #US31 - Como integrante del equipo de trabajo Quiero crear tablero de control con módulos Para continuar con el avance del proyecto integrador.
  + #TK06 - Rol ADMIN.
  + #TK07 - Rol USUARIO.

**Sprint 4**

* #US35 -Como integrante del equipo de trabajo Quiero simular pago con API Mercado Pago Para continuar con el avance del proyecto integrador.
* #US36 -Como integrante del equipo de trabajo Quiero crear un CRUD básico funcional de prod/cliente Para continuar con el avance del proyecto integrador.
* #US34 -Como integrante del equipo de trabajo Quiero mostrar un catálogo Para continuar con el avance del proyecto integrador.
* #US37 -Como integrante del equipo de trabajo Quiero crear una demo Para continuar con el avance del proyecto integrador.
  1. **Conclusión**

Este proyecto ha representado un desafío tanto a nivel personal como grupal. A pesar de los retos que enfrentamos, como la complejidad de las actividades y, en particular, la significativa reducción de integrantes a lo largo del proyecto, consideramos que hemos logrado enfrentar el proyecto con éxito. Inicialmente comenzamos como un equipo de 10 personas, pero debido a diversas circunstancias, terminamos siendo solo 3 integrantes comprometidos hasta el final. Esta reducción de integrantes planteó un desafío adicional que supimos superar con dedicación y esfuerzo.

Es importante destacar que además de la complejidad de las tareas y la reducción de integrantes, contamos con un tiempo limitado para aprender los programas Angular y Django. Sin embargo, a pesar de estas circunstancias adversas, hemos logrado utilizarlos de manera efectiva en el desarrollo del proyecto, demostrando nuestra capacidad de adaptación y compromiso.

Durante este proceso, hemos demostrado un gran compañerismo y una notable capacidad de trabajo en equipo. Valoramos el aprendizaje adquirido y estamos orgullosos del resultado obtenido. A pesar de las limitaciones de tiempo y recursos, hemos aplicado de manera adecuada los conocimientos en Angular y Django, cumpliendo con los objetivos del proyecto.

En resumen, afrontamos diversos desafíos, como la complejidad de las actividades, la reducción significativa de integrantes y el poco tiempo destinado para aprender los programas mencionados. Sin embargo, estamos orgullosos del resultado obtenido y consideramos que hemos superado exitosamente estas dificultades, adquiriendo valiosas habilidades en el proceso de trabajo en equipo.