

# Proyecto: Aerolíneas CBA FLY

## Ficha del documento IEEE-830

Fecha	Revisión	Autor	Verificado dep. Calidad.
29/06/2023		-Rodriguez Moreau, Romina -Neil, Candela Iara -Salvatierra, Walter Facundo -Giannantonio, Elías -Romero, Lucas Emanuel -Colman, Maximiliano -Balduzzi, Lucas Nicolás	

### Índice

#### 1. Introducción

- 1.1. Propósito . . . . .
- 1.2. Alcance . . . . .
- 1.3. Personal involucrado . . . . .
- 1.4. Definiciones, Acrónimos y Abreviaturas . . . . .
- 1.5. Referencias . . . . .
- 1.6. Resumen . . . . .

#### 2. Descripción General

- 2.1. Perspectiva del Producto . . . . .
- 2.2. Características de los Usuarios . . . . .
- 2.3. Restricciones . . . . .

#### 3. Requisitos Específicos

- 3.1. Requerimientos Funcionales . . . . .
- 3.2. Product Backlog (User Stories) . . . . .
- 3.3. Tareas (Tasks) . . . . .
- 3.4. Sprints . . . . .
- 3.5. Conclusiones . . . . .

# 1. Introducción

Este proyecto corresponde a la asignatura Programador Web del ISPC, en el marco de la carrera TSEDWYAD.

Se contemplarán temas de creación de páginas web y SPA que puedan suplir las necesidades y requerimientos de los clientes, en el contexto de un sitio de venta de tickets de aviones. Se especificarán los requerimientos que cumplirá el nuevo módulo de e-commerce que incorporamos a nuestro sitio web.

También se mostrarán todas las tareas necesarias para llegar a completar el sitio web e historias de usuario que ayudarán a crear las funciones que darán valor a nuestro proyecto. El módulo e-commerce incorporado permitirá el acceso a información verídica, respaldada, unificada y almacenada en una base de datos con una estructura de diseño que evitará la duplicación, pérdida, o manipulación no autorizada de la información.

## 1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, para el desarrollo de un sistema de información que permitirá el manejo de una aplicación web en la que los usuarios podrán acceder a información relacionada a vuelos disponibles y a través de la cual se podrán efectuar reservas y compras de pasajes con destino a diferentes ciudades de todo el mundo. Pretendemos que, con el diseño amigable de nuestra página, más gente se anime a viajar y conocer nuevos destinos.

## 1.2 Alcance

Esta especificación de requisitos está dirigida a usuarios interesados en viajar a ciudades puntuales, a través de los vuelos aéreos ofrecidos.

**CBA FLY** es un plataforma digital con la que el usuario podrá:

- Consultar información específica de los vuelos (fechas, horarios, destinos, valores de pasajes, asientos disponibles).
- Registrarse como usuario.
- Modificar sus datos de información personal.
- Reservar pasajes o anular dichas reservas.
- Comprar pasajes con destino único o con destinos múltiples.
- Proceso de compra: Los usuarios podrán seleccionar productos, agregarlos al carrito de compras, modificar el contenido del carrito, seleccionar medios de pago y confirmar dicho pago. Una vez que acepten y finalicen la compra, el proceso se concretará.
- Gestionar sus compras a través de un carrito.
- 

El principal objetivo de esta aplicación será que los clientes puedan visitar destinos no tan comunes y poder satisfacer su curiosidad con nuestro servicio de calidad y confianza.

### 1.3 Personal involucrado

<b>Nombre</b>	Giannantonio, Elías
<b>Rol</b>	Desarrollador - Colaborador
<b>Categoría Profesional</b>	Técnico Superior en Desarrollo Web y Aplicaciones Digitales
<b>Responsabilidad</b>	Diseño / Desarrollo de la aplicación. Frontend y Backend.
<b>Información de contacto</b>	giannantonioelias@gmail.com

<b>Nombre</b>	Colman, Maximiliano
<b>Rol</b>	Desarrollador - Colaborador
<b>Categoría Profesional</b>	Técnico Superior en Desarrollo Web y Aplicaciones Digitales
<b>Responsabilidad</b>	Diseño / Desarrollo de la aplicación. Frontend.
<b>Información de contacto</b>	facumorel1996@gmail.com

<b>Nombre</b>	Rodriguez Moreau, Romina
<b>Rol</b>	Desarrollador - Colaborador
<b>Categoría Profesional</b>	Técnico Superior en Desarrollo Web y Aplicaciones Digitales
<b>Responsabilidad</b>	Diseño / Desarrollo de la aplicación. Backend.
<b>Información de contacto</b>	rominarodriguezmoreau@gmail.com

<b>Nombre</b>	Salvatierra, Walter Facundo
<b>Rol</b>	Desarrollador - <b>Scrum Master</b>
<b>Categoría Profesional</b>	Técnico Superior en Desarrollo Web y Aplicaciones Digitales
<b>Responsabilidad</b>	Diseño / Desarrollo de la aplicación. Llevar a cabo las reuniones, Frontend y Backend.
<b>Información de contacto</b>	fackuu995@gmail.com

<b>Nombre</b>	Neil, Candela Iara
<b>Rol</b>	Desarrollador - Colaborador
<b>Categoría Profesional</b>	Técnico Superior en Desarrollo Web y Aplicaciones Digitales
<b>Responsabilidad</b>	Diseño / Desarrollo de la aplicación. Frontend.
<b>Información de contacto</b>	caianeil27@gmail.com

<b>Nombre</b>	Romero, Lucas Emanuel
<b>Rol</b>	Desarrollador - Colaborador
<b>Categoría Profesional</b>	Técnico Superior en Desarrollo Web y Aplicaciones Digitales
<b>Responsabilidad</b>	Diseño / Desarrollo de la aplicación. Frontend y documentación.
<b>Información de contacto</b>	lucasemmaromero@gmail.com

<b>Nombre</b>	Balduzzi, Lucas Nicolás
---------------	-------------------------

<b>Rol</b>	Desarrollador - Colaborador
<b>Categoría Profesional</b>	Técnico Superior en Desarrollo Web y Aplicaciones Digitales
<b>Responsabilidad</b>	Diseño / Desarrollo de la aplicación. Frontend y Backend.
<b>Información de contacto</b>	lucasnicolasbalduzzi@gmail.com

#### 1.4 Definiciones, acrónimos y abreviaturas

<i>Nombre</i>	<i>Descripción</i>
<b>Usuario</b>	Persona que usará el sistema para gestionar procesos.
<b>SIS-I</b>	Sistema de Información Web para la Gestión de Procesos Administrativos.
<b>ERS</b>	Especificación de Requisitos Software
<b>RF</b>	Requerimiento Funcional
<b>RNF</b>	Requerimiento No Funcional
<b>FTP</b>	Protocolo de Transferencia de Archivos
<b>ISPC</b>	Instituto Superior Politécnico de Córdoba.
<b>TSEDWYAD</b>	Tecnicatura Superior en Desarrollo Web y Aplicaciones Digitales.
<b>US</b>	User Stories o Historias de usuario, son deseos de las personas interesadas en nuestro sitio web que podemos reflejar en las funciones para dejarlos satisfechos.
<b>TK</b>	Tasks o tareas, son los deseos de las personas transformadas en acciones específicas para que la aplicación sea satisfactoria para ellos.

#### 1.5 Referencias

<b>Título del Documento</b>	<b>Referencia</b>
<b>Standard IEEE 830 - 1998</b>	IEEE

#### 1.6 Resumen

El documento está compuesto de tres secciones:

**Sección 1:** Introducción y visión general de la especificación de recursos del sistema.

**Sección 2:** Descripción general del sistema con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

**Sección 3:** Definición detallada de los requisitos que debe satisfacer el sistema.

## 2 Descripción general

### 2.1 Perspectiva del producto

El sistema CBA FLY será un producto diseñado para trabajar en entornos web, lo que permitirá su utilización de forma rápida y eficaz; además se encargará de realizar las siguientes funciones:

- Los usuarios no registrados podrán observar e indagar toda la información de la web CBA FLY y consultar los vuelos disponibles.
- Los usuarios registrados, además de consultar los servicios disponibles, podrán gestionar reservas y comprar pasajes a través de un carrito de compras, abonando con diferentes medios de pago (Efectivo – Tarjeta de crédito – Débito – RapiPago – Pago Fácil)
- El sistema estará diseñado para funcionar en dispositivos móviles y computadoras, y será compatible con los sistemas operativos Windows, Android e iOS.
- La seguridad de los datos estará garantizada mediante el sistema de encriptación proporcionado por el sistema de transacciones de pago incorporado a la plataforma.

### 2.2 Características de los usuarios

<b>Tipo de usuario</b>	<b>Administrador</b>
<b>Formación</b>	Manejo de herramientas informáticas.
<b>Actividades</b>	Control y manejo del sistema en general.

<b>Tipo de usuario</b>	<b>Usuario no registrado</b>
<b>Formación</b>	Manejo de entornos web.
<b>Actividades</b>	Observará el funcionamiento de nuestra aplicación para ver si es de su agrado.

<b>Tipo de usuario</b>	<b>Usuario registrado</b>
<b>Formación</b>	Manejo del sitio con conocimientos básicos de herramientas informáticas.
<b>Actividades</b>	Gestionará reservas - Comprará pasajes – Cambiará sus datos - Usará el carrito de compras.

### 2.3 Restricciones

- Interfaz para ser usada solamente con **conexión a internet**.
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño e implementación sencilla.
- FrontEnd: Utilizará **Angular** para la construcción de la aplicación, **CSS** y **Bootstrap** para los estilos, **Javacript** y **Typescript** para todas las funciones y conexiones al servidor.
- Base de datos: **MySQL**.

- BackEnd: **Python** con el framework **Django** para la creación de clases, funciones y el traer información de la base de datos.
- VCS (sistema de control de versiones): **Git**.
- Nuestro repositorio estará siempre subido en **GitHub** para consultarlo cuando sea necesario.

### 3 Requisitos específicos

#### 3.1 Requerimientos Funcionales y No Funcionales

##### Requerimiento RF01

- Nombre del Requerimiento: **Autenticación del administrador y los usuarios.**
- Características: Los usuarios y el administrador deberán identificarse para acceder al sistema.
- Descripción del Requerimiento: El sistema podrá ser consultado por cualquier usuario y el administrador dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.
- Prioridad del Requerimiento: Alta

##### Requerimiento RF02

- Nombre del Requerimiento: **Registrar usuario.**
- Características Los usuarios tendrán que registrarse en el sistema para comprar vuelos y administrar el carrito de compras.
- Descripción del Requerimiento: El usuario proporcionará sus datos personales (nombre, correo electrónico, teléfono, etc.) solicitados para quedar registrado en el sistema, posteriormente poder iniciar sesión y poder efectuar transacciones de forma adecuada.
- Prioridad del Requerimiento Alta.

##### Requerimiento RF03

- Nombre del Requerimiento: **Modificar Información.**
- Características: El sistema permitirá al Artista, modificar la información que se ofrece en la página y en las novedades.
- Descripción del Requerimiento: Permite al administrador modificar la información proporcionada en las diferentes secciones de la página, como así también la actualización de las novedades, según su nivel de accesibilidad.
- Prioridad del Requerimiento: Alta.

##### Requerimiento RF04

- Nombre del Requerimiento: **Visualización del catálogo de vuelos.**
- Características: El sistema ofrecerá al usuario información sobre todos los vuelos disponibles.
- Descripción del Requerimiento: Muestra información general de los vuelos referida a sus precios, disponibilidad de asientos, fechas de ida y vuelta, origen y destino.
- Prioridad del Requerimiento Alta.

##### Requerimiento RF05

- Nombre del Requerimiento: **Busqueda de vuelos.**
- Características: El sistema tendrá la opción de buscar vuelos a destinos específicos.

- Descripción del Requerimiento: Usando una barra de búsqueda, se podrán filtrar entre varios vuelos para que el usuario pueda encontrar el que esté buscando.
- Prioridad del Requerimiento: Alta.

#### Requerimiento RF06

- Nombre del Requerimiento: **Compras en el carrito.**
- Características: Añadir los vuelos al carrito de compras.
- Descripción del Requerimiento: El usuario podrá elegir los destinos que más le interese, agregarlos al carrito, y luego podrá gestionar su pago en el carrito, eligiendo el método de pago de su preferencia.
- Prioridad del Requerimiento Alta.

## **Requerimientos No Funcionales**

#### Requerimiento RNF01

- Nombre del Requerimiento: **Interfaz del Sistema.**
- Características: El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
- Descripción del Requerimiento: El sistema debe tener una interfaz de uso intuitiva, sencilla, accesible y atractiva.
- Prioridad del Requerimiento: Alta.

#### Requerimiento RNF02

- Nombre del Requerimiento: **Desempeño.**
- Características: El sistema garantizará a los usuarios un desempeño en cuanto a los datos almacenados en el sistema ofreciéndole una confiabilidad a esta misma.
- Descripción del Requerimiento: Garantizar el desempeño del sistema informático a los diferentes usuarios. La información almacenada podrá ser consultada y actualizada permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
- Prioridad del Requerimiento: Alta.

#### Requerimiento RNF03

- Nombre del Requerimiento: **Confiabilidad continúa del sistema.**
- Características: El sistema tendrá que estar en funcionamiento las 24 horas los 7 días de la semana. Ya que es una página web diseñada para la carga de datos, comunicación entre usuarios y e-commerce, pensando en la satisfacción del usuario que operará cuando lo desee.
- Descripción del Requerimiento: La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes.
- Prioridad del Requerimiento: Alta.

#### Requerimiento RNF04

- Nombre del Requerimiento: **Seguridad en Información.**
- Características: El sistema garantizará a los usuarios una seguridad en cuanto a la información que se procede en el sistema, en atención a la privacidad y protección de la información atendiendo al marco jurídico imperante y hacer un uso adecuado de la información de todos los participantes del sitio.
- Descripción del Requerimiento: Garantizar la seguridad del sistema con respecto a la información y datos sensibles como contraseñas.

-Prioridad del Requerimiento: Alta.

## 3.2 Product Backlog

**#US 01**

Como usuario quiero una aplicación web que me brinda información sobre vuelos, destinos, precios, formas de pago y fechas disponibles.

**#US 02**

Como usuario quiero que la aplicación me brinde canales de comunicación para contactar a CBA FLY por vía telefónica, o correo electrónico.

**#US 03**

Como usuario quiero que el sitio me permita registrarme.

**#US 04**

Como usuario registrado quiero poder acceder a hacer reservas y gestionar las compras de pasajes.

**#US 05**

Como usuario registrado quiero recibir las novedades y ofertas promocionales a través de un correo electrónico (newsletter).

**#US 06**

Como cliente quiero que la aplicación me permita cambiar algún pasaje o cancelarlo si fuera necesario.

**#US 07**

Como cliente quiero que la aplicación web me permita abonar los pasajes comprados con tarjeta de débito, crédito, RapiPago y Pago Fácil.

**#US 08**

Como usuario quiero acceder a la edición de mi perfil para poder modificar mis datos.

## 3.3 Tasks

**#TK 01** - Rediseño de la página Inicio: reemplazo de carrusel.

**#TK 02** - Reemplazamos "modo oscuro" por carrito de compras.

**#TK 03** - Solucionar responsive de "Sobre Nosotros".

**#TK 04** - Crear un Diagrama de Clases y Casos de Uso para facilitar el modelado en POO.

**#TK 05** - Actualizar el Diseño Físico (el script legible de la DB en MySQL).

**#TK 06** - Actualizar el Diseño Lógico (consta de la Normalización y el Modelo Relacional).

**#TK 07** - Subir en la carpeta documentación las tablas actualizadas de la base de datos.

**#TK 08** - Actualizar el Diseño conceptual al que ya hemos visto como DER: Diagrama Entidad-Relación.

**#TK 09** - Actualización de documentación IEE830.

**#TK 10** - Revisión y actualización de diagrama UML.



- #TK 11** - Subir documentación faltante.
- #TK 12** - Solucionar bug con selector de filtros "bootstrap-select".
- #TK 13** - Solucionar bug con el Swiper/Slider: funciona solo al cargar el sitio.
- #TK 14** - Solucionar bug con el boton de "ver contraseña" en login/registro.
- #TK 15** - Cambiar vistas de los botones del Nav.
- #TK 16** - Corrección del problema "Angular/core" Version.
- #TK 17** - Construir un Frontend dinámico.
- #TK 18** - Crear servicios que consuman un API Rest de prueba.
- #TK 19** - Inyectar a los componentes los correspondientes servicios.
- #TK 20** - Incorporar la programación reactiva (asincrónica).
- #TK 21** - Establecer el modelo de base de datos del primer Sprint en Django usando clases.
- #TK 22** - Establecer el sistema de usuarios en Django-Admin (registro, login y roles).
- #TK 23** - Crear un ABM de productos/servicios utilizando el panel de control de Django.
- #TK 24** - Solucionar bug: Bootstrap select no se muestra correctamente al cambiar de sección.
- #TK 25** - Revisión y corrección de models.py (Agregar clases meta a cada tabla).
- #TK 26** - Implementacion de clases del Models.py.
- #TK 27** - Crear Formulario Reactivo de Contacto.
- #TK 28** - Crear Formulario Reactivo de Login y Registro.
- #TK 29** - Conexión con el backend para el registro e inicio de sesión.
- #TK 30** - Implementar elToken validación del login con JWT.
- #TK 31** - Crear el Formulario de carrito de compra.
- #TK 32** - Corregir problema con el Formulario de contacto.
- #TK 33** - Hacer un Registro/Alta de al menos 3 productos.
- #TK 34** - Creacion de usuario y admin.
- #TK 35** - Simular Pago con API Mercado Pago o similar.
- #TK 36** - Realizar una Demo final en video con todos los participantes del grupo.

## 3.4 Sprints

<b>Sprint N°</b>	<b>00</b>
<b>Sprint Backlog</b>	<ul style="list-style-type: none"> <li>● Definir <b>Scrum Master inicial</b> y registrarse cada integrante faltante dentro del repositorio a trabajar.</li> <li>● Revisar <b>documentación IEEE830</b> si está completa y retomar desde allí para esta nueva etapa seguir documentando los avances.</li> <li>● Revisar la estructura web Semántica, responsive (RWD)</li> <li>● Definir <b>requerimientos</b> para el nuevo módulo a desarrollar e-commerce.</li> <li>● Plantear <b>Historias de Usuarios y Tareas</b> dependientes de las US para incorporarlas en el repositorio remoto GitHub. (<b>Issues y Milestones</b>)</li> <li>● Definir tareas dentro de las Historias de Usuario (GitHub).</li> </ul>
<b>Responsabilidades</b>	<p><b>Scrum Master:</b></p> <ul style="list-style-type: none"> <li>● Generar un nuevo repositorio con su cuenta de GitHub.</li> <li>● Agregar a los miembros de su equipo.</li> <li>● Registrar en la wiki, las reuniones (meetings) y toda la información pertinente.</li> <li>● Crear Project estilo Kanban con incorporación de Historias de Usuarios, tareas, e incidencias.</li> </ul> <p><b>Colaboradores:</b></p> <ul style="list-style-type: none"> <li>● Crear Branch individual.</li> <li>● Realizar las issues y tareas del sprint.</li> <li>● Completar la documentación IEEE830.</li> </ul>
<b>Calendario</b>	<p><i>Fecha Inicio = 10/04/2023 -</i>  <i>Fecha de Fin = 21/04/2023</i></p>

<b>N° de sprint</b>	<b>01</b>
<b>Sprint Backlog</b>	<ul style="list-style-type: none"> <li>●</li> <li>● Frontend:             <ul style="list-style-type: none"> <li>● Crear una SPA con los módulos y componentes</li> </ul> </li> </ul>

	<p>correspondientes en Angular y sistema de ruteo (routing) en Angular</p> <ul style="list-style-type: none"> <li>• Crear un sistema de ruteo con Angular.</li> <li>•</li> <li>● Backend - Subir en la carpeta documentación las tablas actualizadas de la base de datos: <ul style="list-style-type: none"> <li>• Diseño conceptual al que ya hemos visto como DER: Diagrama Entidad-Relación.</li> <li>• Diseño Lógico que consta de la Normalización y el Modelo Relacional.</li> <li>• Diseño Físico que es el script legible de la DB en MySQL.</li> <li>• Crear un Diagrama de Clases y Casos de Uso para facilitar el modelado en POO.</li> </ul> </li> <li>● Gestión del proyecto: <ul style="list-style-type: none"> <li>• Crear branches individuales para cada miembro del equipo.</li> <li>• Actualizar la Wiki</li> <li>• Corregir el documento IEEE - 830</li> </ul> </li> </ul>
<b>Responsabilidades</b>	<p><b>Scrum Master:</b></p> <ul style="list-style-type: none"> <li>• Registrar en la wiki, las reuniones (meetings) y toda la información pertinente.</li> <li>• Incorporar Historias de Usuarios, tareas, e incidencias en estilo Kanban.</li> </ul> <p><b>Colaboradores:</b></p> <ul style="list-style-type: none"> <li>• Realizar las issues y tareas del sprint.</li> </ul>
<b>Calendario</b>	<p><i>Fecha Inicio = 24/04/2023 -</i>  <i>Fecha de Fin = 05/05/2023</i></p>

N° de sprint	02
Sprint Backlog	<ul style="list-style-type: none"> <li>● Frontend: <ul style="list-style-type: none"> <li>• Crear servicios que consuman un API Rest de prueba. Para realizar las peticiones http, utilizar el servicio de HttpClient.</li> <li>• Inyectar a los componentes los correspondientes servicios a fin de que los componentes puedan acceder a los datos y mostrarlos en pantalla. Incluye enlace de datos (Data binding).</li> <li>• Incorporar la programación reactiva (asincrónica) mediante observables para la comunicación entre servicios y componentes.</li> </ul> </li> <li>● Backend: <ul style="list-style-type: none"> <li>• Establecer el modelo de base de datos del primer sprint en Django usando clases.</li> <li>• Establecer el sistema de usuarios en Django-Admin (registro, login y roles)</li> <li>• Crear un ABM (CRUD) de productos/servicios utilizando el panel de control de Django.</li> </ul> </li> <li>● Gestión del proyecto: <ul style="list-style-type: none"> <li>• Actualizar la Wiki</li> <li>• Corregir el documento IEEE - 830</li> </ul> </li> </ul>
Responsabilidades	<p><b>Scrum Master:</b></p> <ul style="list-style-type: none"> <li>• Registrar en la wiki, las reuniones (meetings) y toda la información pertinente.</li> <li>• Incorporar Historias de Usuarios, tareas, e incidencias en estilo Kanban.</li> </ul> <p><b>Colaboradores:</b></p> <ul style="list-style-type: none"> <li>• Realizar las issues y tareas del sprint.</li> </ul>
Calendario	<p><i>Fecha Inicio = 08/05/2023 -</i>  <i>Fecha de Fin = 21/05/2023</i></p>

N° de sprint	03
Sprint Backlog	<ul style="list-style-type: none"> <li>•</li> <li>● Frontend:             <ul style="list-style-type: none"> <li>• Incorporar formularios reactivos de <b>Registro y Login</b> para la autenticación y formularios para la manipulación (<b>Alta</b>) de <b>servicios/productos/suscripción</b>, según el caso con sus respectivas validaciones.</li> <li>• Conexión con el backend para el registro e inicio de sesión.</li> <li>• Conexión con backend para almacenar los datos de los productos/servicios y detalles de producto/servicios en la DB a través del modelo en Django.</li> <li>• <b>Opcional:</b> Agregar rutas protegidas a la navegación en base al usuario autenticado/autorizado. Ej. para acceder al tablero de control o bien, para acceder al registro de productos/servicios.</li> </ul> </li> <li>•</li> <li>● Backend, definición de la Api Rest y los endpoints necesarios para las funcionalidades de:             <ul style="list-style-type: none"> <li>• Registro de usuarios.</li> <li>• Login de usuarios.</li> <li>• Registro / Alta de al menos 3 productos/servicios/suscripción ( solo desde el rol Administrador, mediante su panel de control).</li> <li>• Tablero de control (Gestión) o Dashboard con módulos planteados en la maquetación del FrontEnd según proyecto:</li> </ul> </li> <li>• <b>El rol ADMIN</b> pueda ver la lista de usuarios registrados y el listado de los productos/servicios/suscripción según corresponda.</li> <li>• <b>El rol USUARIO</b> registrado solo que pueda ver su Dashboard para editar su perfil.             <ul style="list-style-type: none"> <li>• Los datos serán consumidos desde la Base de datos MySQL.</li> <li>• <b>OPCIONAL:</b> Token validación del login con JWT.</li> </ul> </li> <li>● Gestión del proyecto:             <ul style="list-style-type: none"> <li>• Subir al repo grupal en GitHub, mediante comandos GIT desde el local, pueden crearla en una branch por desarrollador para que cada uno tenga una copia, o directamente crear una branch <b>FEATURE</b>, para actualizar los cambios</li> </ul> </li> </ul>

	<p>que no se encuentran aún en condiciones de incorporarse a la branch <b>DEVELOP</b>.</p> <p>En GitHub, mediante la sección WIKI deben reflejar los avances individuales y de equipo en relación a Scrum (Ceremonias, roles, historias de usuarios, etc). Actualizar IEEE830.</p> <ul style="list-style-type: none"> <li>• <b>“FEATURE”</b> son las branches en las que se trabaja la mayor parte del tiempo. Cuando no se utilizan más se borran.</li> <li>• <b>“DEVELOP”</b> es la branch donde se mantiene la versión de desarrollo más estable y reciente. La que es candidata para pasar a ser la principal (<b>MAIN</b>).</li> <li>• La branch <b>“MAIN”</b> o <b>“MASTER”</b> es la que suele usarse para mantener las versiones estables (productivas) de nuestros proyectos. Listas para salir a producción. Se supone que en cada cierre de Sprint, queda lo solicitado en esa condición, por lo que: <ul style="list-style-type: none"> <li>• <i>Para el cierre de cada SPRINT si deberá realizarse un MERGE a la rama <b>MAIN/MASTER</b> como constancia que está completo dicho Sprint para la <b>Review</b> (en producción)***</i></li> </ul> </li> </ul> <p><i>Estas son algunas CONVENCIONES más usadas para trabajar con branches en proyectos de desarrollo de software.</i></p> <p><b>*** NOTA:</b> para agilizar la vista del proyecto y simular una presentación de una "Demo" será necesario que el grupo completo realice una presentación del producto (MVP) con las funcionalidades incorporadas hasta el momento del cierre de Sprint.</p> <ul style="list-style-type: none"> <li>• Actualizar la Wiki</li> <li>• Corregir el documento IEEE - 830</li> </ul>
<b>Responsabilidades</b>	<p><b>Scrum Master:</b></p> <ul style="list-style-type: none"> <li>• Registrar en la wiki, las reuniones (meetings) y toda la información pertinente.</li> <li>• Incorporar Historias de Usuarios, tareas, e incidencias en estilo Kanban.</li> </ul> <p><b>Colaboradores:</b></p> <ul style="list-style-type: none"> <li>• Realizar las issues y tareas del sprint.</li> </ul>
<b>Calendario</b>	<p><i>Fecha Inicio = 22/05/2023 -</i>  <i>Fecha de Fin = 04/06/2023</i></p>

N° de sprint	04
Sprint Backlog	<ul style="list-style-type: none"> <li>•</li> <li>● Frontend + Backend integrado:             <ul style="list-style-type: none"> <li>• <i>Tablero de control para usuarios:</i> Mostrar catálogo (mínimo 3 productos o servicios o suscripción). Seleccionar para comprar. Simular Pago con API Mercado Pago o similar</li> <li>• <i>Tablero de control administrador CRUD</i> básico funcional de producto y cliente.</li> <li>• Incorporar la programación reactiva (asincrónica) mediante observables para la comunicación entre servicios y componentes.</li> </ul> </li> <li>● Gestión del proyecto:             <ul style="list-style-type: none"> <li>• Subir al repo grupal en GitHub, mediante comandos GIT desde el local, pueden crearla en una branch por desarrollador para que cada uno tenga una copia, o directamente crear una branch <b>FEATURE</b>, para actualizar los cambios que no se encuentran aún en condiciones de incorporarse a la branch <b>DEVELOP</b>.</li> <li>• En GitHub, mediante la sección WIKI deben reflejar los avances individuales y de equipo en relación a Scrum (Ceremonias, roles, historias de usuarios, etc). Actualizar IEEE830.</li> </ul> </li> <li>● “<b>FEATURE</b>” son las branches en las que se trabaja la mayor parte del tiempo. Cuando no se utilizan más se borran.</li> <li>● “<b>DEVELOP</b>” es la branch donde se mantiene la versión de desarrollo más estable y reciente. La que es candidata para pasar a ser la principal (<b>MAIN</b>).</li> <li>● La branch “<b>MAIN</b>” o “<b>MASTER</b>” es la que suele usarse para mantener las versiones estables (productivas) de nuestros proyectos. Listas para salir a producción. Se supone que en cada cierre de Sprint, queda lo solicitado en esa condición, por lo que:             <ul style="list-style-type: none"> <li>• <i>Para el cierre de cada SPRINT si deberá realizarse un MERGE a la rama <b>MAIN/MASTER</b> como constancia que está completo dicho Sprint para la <b>Review</b> (en producción)***</i></li> </ul> </li> </ul> <p><i>Estas son algunas CONVENCIONES más usadas para</i></p>

	<p><i>trabajar con branches en proyectos de desarrollo de software.</i></p> <ul style="list-style-type: none"> <li>● <b>*** NOTA:</b> para agilizar la vista del proyecto y simular una presentación de una "Demo" será necesario que el grupo completo realice una presentación del producto (MVP) con las funcionalidades incorporadas hasta el momento del cierre de Sprint. <ul style="list-style-type: none"> <li>● Actualizar la Wiki</li> <li>● Corregir el documento IEEE – 830</li> <li>● Actualizado todo el proyecto en GitHub como respaldo</li> <li>● <i>Demo final</i> en video con participación de todo el equipo describiendo todas las funcionalidades. (quien no explique en el mismo alguna de las funcionalidades/código/gestión del proyecto, no será tenido en cuenta para calificación)</li> <li>● OPCIONAL: Sitio publicado en Servidor</li> </ul> </li> </ul>
<b>Responsabilidades</b>	<p><b>Scrum Master:</b></p> <ul style="list-style-type: none"> <li>● Registrar en la wiki, las reuniones (meetings) y toda la información pertinente.</li> <li>● Incorporar Historias de Usuarios, tareas, e incidencias en estilo Kanban.</li> </ul> <p><b>Colaboradores:</b></p> <ul style="list-style-type: none"> <li>● Realizar las issues y tareas del sprint.</li> </ul>
<b>Calendario</b>	<p><i>Fecha Inicio = 05/06/2023 -</i>  <i>Fecha de Fin = 18/06/2023</i></p>



## **3.5 Conclusiones**

- 1- Desvinculaciones de integrantes de equipos. Empezamos siendo diez personas, se fueron dos, entró una nueva, se fue alguien más y cerca del Sprint 4 se va otra persona más.
- 2- Creemos que el tiempo de entrega fue adecuado, pero la falta de material de estudio provisto por la plataforma nos dificultó un poco más el trabajo.
- 3- Hasta ahora tenemos algunos bugs pero nada que arruine la aplicación ya que son más referidos a Git.
- 4- Los conocimientos de cada miembro del grupo podían llegar a ser bastantes dispares, pero pudimos dividir bien las tareas para que todos podamos hacer nuestra parte.
- 5- La metodología Scrum sigue siendo un poco difícil de llevar a cabo pero hemos aprendido a apreciar el poder entregar productos escalables y viables.