Práctica Profesional I 1 2022

# Documento para 1° evaluación: Funcionalidades Definitivas Producto 20%

### Grupo: EZ Solutions

### Nombre de Producto: Acturus Finance

| **Nº** | **Funcionalidad Aplicativo** | **Descripción** | **Impacto** | **Esfuerzo** |
| --- | --- | --- | --- | --- |
| 1 | Autentificación eje login | Autentificación que permitirá al usuario ingresar a la aplicación, aquí manejamos el login y la registración. | Un control de seguridad moderno y eficiente para resguardar tu cuenta personal | 10 horas de desarrollo |
| 2 | Verificación de identidad | Al tratarse de una aplicación con información sensible se debe de comprobar la identidad de la persona | Un mejor parámetro de seguridad para evitar estafas o robo de identidad | 7 horas de desarrollo |
| 3 | Patrón de diseño MVC | Ya que es una aplicación grande, para actualizaciones futuras y fácil manejo de información se recomienda dividir el sistema por capas | Mejora el flujo de desarrollo: frontend, backend y blockchain | 16 horas de desarrollo |
| 4 | Sistema de movimientos | El usuario podrá ver sus movimientos, sus transacciones. | Facilita al usuario un feedback de su status en la plataforma y estados de cuenta | 10 horas de desarrollo |
| 5 | Diagrama de arquitectura de alto nivel | Esto permitirá apreciar de mejor manera la información de los procesos. | Mejora ver de manera general, y definir cada una de las partes mediante diagramas. | 10 horas de desarrollo |
| 6 | Diagrama de flujo/UML de aplicación | Un diagrama de flujo sirve para ubicar mejor el orden de los procesos o algoritmos de un sistema grande. | El diagrama de flujo facilita ver el comportamientos de las pantallas | 10 horas de desarrollo |
| 7 | Diseño Login | Diseño del mockup del login, los detalles dependen del arte de la empresa. | Facilita la visualización de los campos para ingresar los datos de autenticación | 5 horas de desarrollo |
| 8 | Desarrollo de Login | El sistema tendrá un tipo de login mediante una wallet. | Facilita el ingreso a la aplicación a través de campos visibles y fáciles de identificar | 6 horas de desarrollo |
| 9 | Diseño Registro | El registro de es el registro de la wallet a conectarse | Facilita la visualización de los campos necesarios para la creación de una cuenta nueva | 5 horas de desarrollo |
| 10 | Desarrollo de Registro | El Registro dependerá de la wallet | El Registro mejora la captura de datos para la fácil creación de una cuenta | 6 horas de desarrollo |
| 11 | Diseño recuperacion contraseña | El Mockup la pantalla para recuperar contraseñas dependerá de la wallet | Facilita la visualización de los campos para el desarrollo de la pantalla recuperación de contraseña. | 5 horas de desarrollo |
| 12 | Desarrollo recuperación contraseña | Para recuperar contraseñas dependerá de la wallet | Fácil recuperación de información, opciones por si hay pérdida de contraseña. | 6 horas de desarrollo |
| 13 | Diseño pantalla home | El Mockup del diseño de pantalla home. | Facilita la visualización de los campos para el desarrollo de la pantalla home | 5 horas de desarrollo |
| 14 | Desarrollo pantalla home | La pantalla home estará conectada con el backend para proporcionarle información | Ahorro de tiempo para ver información importante del usuario | 6 horas de desarrollo |
| 15 | Diseño perfil usuario | El Mockup del sistema diseño perfil usuario. | Mejora la visualización de los datos personales del cliente | 5 horas de desarrollo |
| 16 | Desarrollo perfil usuario | Esta pantalla debe contener la información personal de cada usuario. Servirá de fácil acceso |  | 7 horas de desarrollo |
| 17 | Diseño editar perfil usuario | El Mockup del sistema edita perfil usuario. | Facilita la visualización de los campos para el desarrollo de la pantalla de edición para el perfil usuario | 5 horas de desarrollo |
| 18 | Desarrollo editar perfil usuario | El Usuario podrá cambiar su información. |  | 7 horas de desarrollo |
| 19 | Diseño pantalla transacciones | El Mockup del sistema pantalla transacciones. | Facilita la visualización de los campos para el desarrollo del sistema de transacciones. | 5 horas de desarrollo |
| 20 | Desarrollo pantalla transacciones | Esta pantalla consiste en la creación de transacciones, realizar transacciones de cuentas | Método fácil y seguro para enviar dinero sin pasar por un medio como el banco, pagó persona a persona. | 10 horas de desarrollo |
| 21 | Diseño pantalla verificación transacción | El Mockup del sistema verificación transacción | Facilita la visualización de los campos para el desarrollo la pantalla de verificación de transacciones. | 5 horas de desarrollo |
| 22 | Desarrollo pantalla verificación transacción | Se tendrá que realizar una verificación previa | Seguridad para el usuario a la hora de realizar una transacción. | 6 horas de desarrollo |
| 23 | Diseño historial de transacciones. | El Mockup del sistema registro | Facilita la visualización de los campos para el desarrollo la pantalla historial de transacciones. | 5 horas de desarrollo |
| 24 | Desarrollo histórico de transacciones. | El frontend hará un request para poder traer el historial de las transacciones | Facilidad para llevar un registro de movimientos de gastos y ganancias. | 6 horas de desarrollo |
| 25 | Diseño detalles transacción histórica | El Mockup del sistema registro | Facilita la visualización de los campos para el desarrollo la pantalla de detalles de transacción histórica. | 5 horas de desarrollo |
| 26 | Desarrollo detalles de la transacción. | El frontend podrá tener acceso a la información de las transacciones | Es un punto importante contaré una información precisa de los detalles de cada transacción. | 6 horas de desarrollo |
| 27 | Diseño ingreso de favoritos | El Mockup del sistema registro | Es de impacto para que el usuario no se desubique donde están las opciones. | 5 horas de desarrollo |
| 28 | Desarrollo ingreso de favoritos. | Se refiere a los usuarios como favoritos. | Esta funcionalidad permite tener información a la mano. | 6 horas de desarrollo |
| 29 | Conexión directa al blockchain | El blockchain se encargará de brindar información y algoritmos, que servirán de ayuda para un mejor desarrollo. | Resuelve la parte de seguridad y darle valor a una moneda digital. También la versatilidad | 10 horas de desarrollo |
| 30 | Lógica de transacciones | Realizará una serie de validaciones, y poder generar el debito | Sirve realizar los débitos personales, a cuentas externas de manera segura y rápida | 10 horas de desarrollo |
| 31 | Flujos de integración continua | Configuración del ambiente para una facilidad del deploy | Resuelve la transacción de desarrollo a producción | 5 horas de desarrollo |
| 32 | Tecnología escalable | Fácil uso de integracion de tecnologias actuales que permitan un fácil mantenimiento | Resuelve la escalabilidad del proyecto, así agregando más funcionalidades a futuro | 10 horas de desarrollo |
| 33 | Sistema multiplataforma | La plataforma se podrá visualizar en diferentes dispositivos y OS | Tiene un gran impacto, al existir multiples OS, dispositivos, esta se adaptara, a los más recientes | 20 horas de desarrollo |
| 34 | Formularios para captura de datos | Captura información del sistema por parte del usuario. | Las entradas de datos son muy importantes, ya que gracias a ello, se puede recibir, datos, enviar información precisa | 5 horas de desarrollo |

## Generalidades de la entrega

Básicamente el contenido de este documento es el mismo de la entrega de la semana 8. Solo tendría algunas observaciones:

* Cada miembro del grupo será responsable de funcionalidades completas
* Un buen criterio para acotar una funcionalidad es que se pueda desarrollar en las 15 semanas de PP2, es decir: 15 semanas \* 7.5 horas hombre/semana =***105 horas/ hombre*** o menos.

Revisen cada funcionalidad definida y agruparse en unos de los 5 grupos de funcionalidades expuestos en clase.

**Funcionalidades para lograr las Gains:** Funcionalidades que resuelven (algunos) de los problemas encontrados o hace al usuario gozar de las ventajas del proceso electrónico de datos

**Funcionalidades Back Office:**  (las no-funcionales del video) Las funcionalidades BackOffice (FBO) son acciones que hacen los programas, pero son transparentes al usuario. Por ejemplo, su programa sugiere un valor de lo que se les ocurra, pero para hacer esa sugerencia ejecuta un sofisticado algoritmo de inteligencia artificial. Ese cálculo es una FBO.

**Funcionalidades estándar en la industria:**Son todas aquellas funcionalidades que debe llevar toda aplicación por ser estándar “de facto 2 en la industria en que estamos(i.e. informática).Ejemplos serían:

* El despliegue de calendarios para definir una fecha
* Un pequeño comentario guía al posicionarnos en un campo cualquiera
* Logs de auditoría
* Mecanismo para recuperar la contraseña

Estas funcionalidades sólo se definen una vez, de manera general para todo el aplicativo  
**Funcionalidades Interacción otros sistemas:** Debemos considerar las funcionalidades necesarias para comunicarse con aplicativos externos, ejemplo prepara los paquetes de datos necesarios al enviar un pago con tarjeta. Etc. Como no sabemos que vamos a necesitar pasar como parámetro al aplicativo externo, bastará con escribir “preparar datos para aplicación de pago con tarjeta” y, al revés “interpretar datos recibidos de aplicación pago con tarjeta” (para el ejemplo que mencionamos)

**Funcionalidades de “trabajo de calidad” o “milla extra”:** Son aquellas funcionalidades que nadie ha pedido pero que nosotros las consideramos necesarias para hacer un trabajo de calidad, pues nuestra imagen profesional está de por medio.

Por ejemplo:

* Mensajes de error orientadores
* Barras de avance del proceso
* Implementación de “teclas de pánico”
* Reportes e indicadores resultado del log de auditoría

## Impacto y esfuerzo, insumos para encontrar el MPV

### Impacto:

Establezcan un criterio de impacto no basado en lo que la funcionalidad es importante o no para la aplicación. El criterio de impacto debería ser aquello que la aplicación resolverá y que impacta en el auxilio que da la aplicación para realzar el trabajo del usuario: más fácil, más control sobre el proceso, más facilidad para visualización, ahorro de tiempo, evitar cálculos engorrosos etc.

Del impacto que causen dependerá que les siga ayudando quién les ayuda, reclutar nuevas personas dispuestas a ayudar para desarrollar un producto mejor o lograr que alguien invierta dinero en su producto. Recuerden aquella frase de Dale Carnegie:

"Personalmente, soy muy aficionado a las fresas con crema, pero he encontrado que, por alguna extraña razón, los peces prefieren los gusanos. Así que cuando fui a pescar, no pensé en lo que yo quería. Pensé en lo que ellos querían: puse gusanos en el anzuelo, no fresas con crema."

### Esfuerzo:

* Deberían procurar que las funcionalidades que vayan a estar ubicadas en el MPV suman 105 horas/hombre \* # de integrantes del grupo. Ej. En un grupo de 5 deberán sumar 105 \* 5= 525 horas
* Sugiero que utilicen alguna técnica estándar para evaluar el esfuerzo requerido (en horas) para desarrollar una funcionalidad. Ejemplo el “Planning Poker''. Está descrito en <https://samuelcasanova.com/2016/01/estimacion-agil-con-la-tecnica-planning-poker/> y otros muchos sitios mas. He subido una presentación más.

En ambos casos, Impactó – Esfuerzo, esperaría que ustedes diseñen sus propias escalas de evaluación, me lo comuniquen en el presente documento y las apliquen.

Esperaría también que el trabajo fuera en común y todo el grupo estuviera enterado del trabajo de todos.

## Criterios de calidad para Evaluación FUNCIONALIDADES DEFINITIVAS

A continuación, les detallo los criterios que ensayamos aplicar para evaluar el documento presentado:

| **N°** | **Sección** | **Indicador** | **Nivel excelente** | **Nivel**  **Promedio** | **Nivel mínimo aceptable** | **No aceptable** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | ***Funcionalidad Aplicativo*** | Nombre funcionalidad | Describe en 3 o 4 palabras lo que hace la funcionalidad | N/A | Describe en más de 5 palabras lo que hace la funcionalidad | No se relaciona con lo que hace la funcionalidad o usa palabras de manera equivocada o inexacta. |
| 2 | ***Descripción*** | Texto Descripción | Explica claramente y en detalle lo que hace la funcionalidad, reflejando las apuestas tecnológicas en las que se fundamenta | Explica claramente que hace la funcionalidad pero falta detalle | Explica de manera vaga pero aproximada lo que hace la funcionalidad. | Describe de manera confusa lo que hace la funcionalidad o usa palabras de manera equivocada o inexacta. |
| 3 | ***Impacto*** | Grado de satisfacción del usuario con la funcionalidad. | La escala de evaluación está clara para el usuario y la calificación corresponde al impacto que producirá en su negocio/trabajo | La escala de evaluación está clara para el usuario pero no alcanza a evaluar el impacto que producirá en su negocio/trabajo | La escala de evaluación es vaga y el usuario no alcanza a ver impacto que producirá en su negocio/trabajo | El impacto está evaluado a un criterio que no se refiere a las “ganancias” que el usuario obtendrá de la funcionalidad. |
| 4 | ***Esfuerzo*** | Esfuerzo que deberá invertir quien desarrolle la funcionalidad. Debe ser m  medido en horas e incluye no solo las tareas asociadas al desarrollo de Software sino también tareas de investigación. | Hay una evaluación de esfuerzo a través de una técnica aceptada como Planning Poker o Técnica Delphi1 | Hay una evaluación de esfuerzo a través de alguna técnica aceptada, pero no aplicada en su totalidad | Hay una evaluación de esfuerzo de manera intuitiva, en base a la experiencia del grupo. | No hay método o se produce al azar |

1 <http://www.pmoinformatica.com/2018/08/tecnicas-estimacion-software.html#:~:text=Existen%20diversas%20t%C3%A9cnicas%20de%20estimaci%C3%B3n,puntos%20de%20funci%C3%B3n%20y%20COSMIC>.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*