

1. Introducción

Como parte de los trabajos domiciliarios de la materia Ingeniería de Software 1, se presenta el siguiente proyecto. El mismo consistió en el desarrollo de una aplicación que permita a los usuarios alimentarse de forma saludable, además, se elaboró este documento en donde se incluyen los estándares de código empleados en dicha realización, evidencia de pruebas funcionales, reporte de defectos, entre otros.

Cada sección fue cuidadosamente confeccionada de acuerdo a los que el equipo considero los “mejores” estándares disponibles. Para su elección se investigó tanto en la web como en distintas piezas bibliográficas.

2. Versionado

2.1. Repositorio utilizado

Se utilizó el repositorio Git para almacenar los componentes de software producidos a lo largo de todo el proyecto. Con la ayuda de esta herramienta, se gestionó el cambio de manera paralela. Además, aseguró que siempre se estuviera trabajando a partir de la última versión creada.

Git modela sus datos “como un conjunto de instantáneas de un mini sistema de archivos”¹. Al confirmar un cambio o guardar el estado del proyecto en Git, se realiza una especie de imagen del aspecto de todos los archivos en dicho momento, y se almacena una referencia. Como técnica para mejorar la eficiencia, si los archivos no han sufrido modificaciones, Git no vuelve a almacenar el archivo sino que solamente guarda un puntero archivo anterior idéntico que fue almacenado anteriormente.

La representación de versiones en git se realiza mediante un grafo acíclico dirigido.

- Nodos: Representan las versiones
- Aristas dirigidas: Representa la relación padre-hijo entre versiones
- Nodo raíz: Primera versión
- Hojas: Representan las últimas versiones

Los Elementos de la Configuración de Software (ECS) incluidos en el repositorio durante el desarrollo del proyecto se componen de:

- Programas: código fuente, ejecutable, proyecto de Netbeans.
- Documentos: plantilla de overleaf, pruebas unitarias tomadas en cuenta como documentación al día del proyecto.

2.2. Criterios de versionado

Git es un repositorio distribuido, ya que promueve el uso de ramificaciones en el desarrollo. Cuando hay cambios concurrentes se guardan en el repositorio las versiones concurrentes que

¹<https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>

eventualmente se pueden combinar (operación merge). Su ventaja reside en permitir trabajar offline y tomar decisiones antes de decidir cómo combinar mi trabajo con el de otro (Commit before merge).

Se llama ramificación a la creación de una nueva línea de código a partir de una versión en una línea de código ya existente. Crea espacios virtuales, por ejemplo, si se detecta un defecto en un proyecto, se puede crear una línea de trabajo paralelo para arreglarlo, basada en una versión estable del producto.

Ejemplos de comandos fundamentales:

- `git branch (nombreRama)` : genera una nueva rama.
- `git merge (nombreRama) (nombreRama2)` : combinar el contenido de una rama con otra.
- `git checkout (nombreRama) (nombreRama2)`: cambio de una rama a otra.
- `git tag`: permite agregar tags, muy útil para el versionado.

Ciclo básico de trabajo:

1. Crear repositorio
2. Actualizar área de trabajo: check out
3. Hacer cambios: add, delete, move, copy
4. Examinar cambios: status, diff
5. Combinar con otros cambios: update, revert, resolved
6. Comitear los cambios

Los commits, son comentarios asociados a cambios confeccionados. Por ejemplo, realizar un push al repositorio online, solucionando conflictos, u en otras situaciones. Ayudan a explicitar sobre qué secciones se trabajó, en qué momento y quién realizó dichos trabajos. Según el propio sitio de git² existen dos clases de commits:

Commits (locales): Los commits en Git son locales hasta que no se efectúa la subida al servidor. Estos commits locales pueden ser modificados sin peligro (se les pueden añadir más cambios, actualizar su mensaje o incluso eliminarlos).

Commits (remotos): Cuando se suben cambios al repositorio remoto, se considera que estos entran a formar parte del histórico compartido entre los desarrolladores del proyecto y, por lo tanto, no es buena práctica modificarlos del mismo modo en que se hace cuando los commits son locales.

Para el versionado de la aplicación se utilizó el estándar confeccionado por Tom Preston-Werner, llamado: "The Semantic Versioning Specification".³.

²<https://gist.github.com/juanguhurtado/7a819d4f07619e944b56>

³Sitio web: <http://semver.org/summary>

Un número normal de versión tiene la forma X.Y.Z donde X, Y, y Z son enteros no negativos. X es la versión “major”, Y es la versión “minor”, y Z es la versión “patch”. Cada elemento DEBE incrementarse numéricamente en incrementos de a 1. Por ejemplo: 1.9.0 – 1.10.0 – 1.11.0. Una de las reglas más relevantes de este estándar es que, una vez que una versión fue liberada, queda terminantemente prohibido modificarla y relanzarla, se debe lanzar como una nueva versión con las modificaciones pertinentes.

La Z debe incrementarse cuando se introducen solo arreglos compatibles con la versión anterior. Un arreglo de bug se define como un cambio interno que corrige un comportamiento erróneo. La versión minor Y debe ser incrementada si se introduce una nueva funcionalidad compatible con la versión anterior. Puede ser incrementada si se agrega funcionalidad o arreglos considerables al código privado. Otro punto importante es que debe resetearse a 0 cuando la versión minor es incrementada.

Finalmente, la versión major X debe ser incrementada si hay cualquier cambio no compatible con la versión anterior. Puede incluir cambios de nivel minor y/o patch. Las versiones patch y minor deben ser reseteadas a 0 cuando se incrementa la versión major.

En nuestro proyecto el criterio para el cambio de versiones aplicando el estándar resultó:

- Aumento de Z cuando el cambio es interno a alguna funcionalidad y no representa un cambio significativo ni a nivel de lógica ni a nivel de interfaz. Ejemplos: cambiar el tamaño de una label, cambiar colores de letra, entre otros.

- Aumento de Y cuando se agrega alguna funcionalidad o elemento de interfaz relevante. También cuando se hace algún cambio importante a nivel de lógica. Ejemplos: agregar funcionalidad consulta con profesional, agregar la opción de solicitar planes de alimentación, agregar ventana de ver perfil de usuario, entre otros.

- Aumento de X cuando se llega a una nueva versión estable del sistema.

2.3. Resumen del log de versiones

Principales versiones del sistema:

Versión 0.0.1: Punto de inicio, creación y configuración del proyecto de NetBeans base. Organización de carpetas dentro del repositorio para pautar almacenamiento de los avances.

Versión 0.2.0: En esta versión ya se tienen las principales clases de objetos que modelarán el problema, incluyendo la clase Sistema, la cual está encargada de mediar en la interacción dominio-interfaz.

Versión 1.0.0: Primer versión estable en cuanto al dominio, la interfaz no ha sido incorporada aún.

Versión 1.3.0: Esta versión posee un dominio estable y la ventana principal del sistema a través de la cual se registran las dos clases de personas que pueden ingresar al sistema: profesional y usuario.

- Entrada: Hay al menos 1 usuario ya creado y se muestran todos en la lista izquierda de la ventana principal. Click en cualquiera de los nombres que allí figuran.
- Salida esperada: Se abre menú principal del usuario. A la izquierda de la ventana se tienen las opciones consulta con profesional, solicitar plan de alimentación e ingreso de alimento ingerido.

d) Ingresar al sistema (Profesional)

1) Caso 1:

- Entrada: No hay profesionales creados en el sistema. Lista de profesionales en ventana principal está vacía. Hacer click en lista vacía.
- Salida esperada: Sistema avisa que no hay profesionales creados.

2) Caso 2:

- Entrada: Hay al menos un profesional ya creado y se muestran todos en la lista derecha de la ventana principal. Click en cualquiera de los nombres que allí figuran.
- Salida esperada: Se abre menú principal del profesional. A la derecha de la ventana se tienen las opciones atender consultas pendientes, atender solicitudes de planes de alimentación e ingreso de alimento al sistema.

2. FUNCIONALIDADES (ROL USUARIO)

Pre-condición: Se debe contar con al menos un usuario registrado en el sistema e ingresar al menú principal del usuario mediante uno de ellos.

a) Consulta con profesional

Pre-condición 1: Hacer click en botón consulta con profesionales.

Pre-condición 2: Al menos un profesional registrado en sistema. (No aplica para caso 1)

1) Caso 1:

- Entrada: No hay profesionales creados en el sistema.
- Salida esperada: Sistema avisa que no hay profesionales registrados en el sistema y no permite acceder a la funcionalidad.

2) Caso 2:

- Entrada: No hay conversaciones creadas en el sistema aún que incluyan a ese usuario. Hacer click en icono + para crear una nueva conversación. Elegir un profesional de la lista que se despliega. Escribir y enviar mensaje.
- Salida esperada: Sistema despliega mensaje enviado en pantalla y permite seguir enviando mensajes al mismo profesional y/o a otros (previa selección en la lista ubicada a la izquierda de la pantalla).

3) Caso 3:

- Entrada: Ya existen conversaciones creadas para el usuario logueado. Elegir conversación con alguno de los profesionales de la lista. Escribir mensaje y enviar.
- Salida esperada: Sistema despliega mensaje enviado en pantalla y permite seguir enviando mensajes al mismo profesional y/o a otros (previa selección en la lista ubicada a la izquierda de la pantalla).
NOTA: Se puede iterar sobre esta acción y la salida esperada es siempre igual para cada conversación/profesional según corresponda.

4) Caso 4:

- Entrada: Ya existen conversaciones creadas. Click en botón + para crear nueva conversación. Sistema despliega lista con profesionales a los cuales no se ha contactado aún. Seleccionar uno.
- Salida esperada: Se crea una conversación entre el profesional elegido y el usuario y se puede enviar mensajes al mismo y a los que ya estaban.

b) Plan de alimentación

Pre-condición 1: Hacer click en botón plan de alimentación.

Pre-condición 2: Al menos un profesional registrado en sistema. (No aplica para caso 1)

1) Caso 1:

- Entrada: No hay profesionales creados en el sistema.
- Salida esperada: Sistema avisa que no hay profesionales registrados en el sistema y no permite acceder a la funcionalidad.

2) Caso 2: Pre-condición: Hacer click en botón izquierdo superior (calendario con un signo de adición en el centro).

- Entrada: Seleccionar un profesional del combobox disponible para dichos fines. Click en el ícono inferior (tick).
- Salida esperada: Sistema avisa que el plan ha sido solicitado con éxito.

3) Caso 3: Pre-condición: No hay planes de alimentación creados aún para el usuario logueado.

- Entrada: Hacer click en botón derecho superior (ícono de libreta con lupa)
- Salida esperada: Sistema avisa que no hay planes disponibles e impide acceder a la opción.

4) Caso 4: Pre-condición: Hay al menos un plan de alimentación creado para el usuario logueado.

- Entrada: Hacer click en botón derecho superior (ícono de libreta con lupa). Click en algún ítem de la lista de planes.
- Salida esperada: Sistema despliega nueva pantalla mostrando los detalles del plan de alimentación seleccionado. NOTA: Se puede iterar sobre este caso para testear todos los planes ya creados para el usuario logueado.

c) **Ingresar alimento ingerido** Pre-condición 1: Hacer click en botón ingresar alimento ingerido.

Pre-condición 2: Hay al menos un alimento registrado en el sistema. (No aplica para caso 1)

1) Caso 1:

- Entrada: No hay alimentos registrados en el sistema.
- Salida esperada: Sistema despliega un panel confirmando la ausencia de alimentos y no permite acceder a la funcionalidad hasta que esto sea corregido.

2) Caso 2:

- Entrada: Se selecciona un alimento del combobox destinado a dicho fin. Se elige una fecha para la ingesta y se clickea en el tick para confirmar el registro.
- Salida esperada: Sistema despliega un panel aceptando la ingesta. NOTA: Se puede realizar este caso de manera iterativa, cada vez que se clickea en ingresar alimento ingerido, los campos se resetearán. No hay límite en la cantidad de alimentos ingeridos por un usuario.

3) Caso 3:

- Entrada: Se deja uno o más campos sin llenar.
- Salida esperada: Sistema solicita el ingreso de todos los datos de manera obligatoria, de lo contrario el registro no se realiza. En caso de llenarse los campos restantes se toma el mismo curso normal del caso 2.

3. FUNCIONALIDADES (ROL PROFESIONAL)

Pre-condición: Se debe contar con al menos 1 profesional registrado en el sistema e ingresar al menú principal del profesional mediante uno de ellos.

a) **Consultas pendientes** Pre-condición 1: Hacer click en botón consultas pendientes.

Pre-condición 2: Al menos una consulta realizada por parte de los usuarios. (No aplica para caso 1)

1) Caso 1:

- Entrada: No hay consultas creadas.
- Salida esperada: Sistema avisa que no hay consultas para atender en el sistema y no permite acceder a la funcionalidad.

2) Caso 2:

- Entrada: Se selecciona un usuario de la lista que se encuentra en el lado izquierdo de la pantalla. En el text area desplegado por el sistema se ingresa el mensaje y se clickea en el botón para enviarlo. (flecha ubicada abajo a la derecha)

- Salida esperada: Sistema envía mensaje a usuario (el que había realizado la consulta seleccionada) y muestra un panel confirmando el mensaje. NOTA: Se puede realizar de forma iterativa.
- b) **Planes de alimentación solicitados** Pre-condición 1: Hacer click en botón planes solicitados.
 Pre-condición 2: Al menos una solicitud realizada por parte de un usuario. (No aplica para caso 1) Pre-condición 3: Al menos un alimento ingresado al sistema por parte de algún profesional. (No aplica para caso 2)
 - 1) Caso 1:
 - Entrada: No hay planes de alimentación.
 - Salida esperada: Sistema avisa que no hay solicitudes registradas en el sistema para el profesional logueado y no permite acceder a la funcionalidad hasta que las haya.
 - 2) Caso 2:
 - Entrada: No hay alimentos creados en el sistema.
 - Salida esperada: Sistema informa de la inexistencia de alimentos en el sistema y bloquea el acceso a la funcionalidad hasta que los haya.
 - 3) Caso 3:
 - Entrada: Se elige un usuario de la lista desplegada, la cual incluye a todos los usuarios que hayan solicitado la realización del mismo, y no hayan obtenido respuesta aún. El sistema despliega la información del usuario. Clickear en elaborar plan. Datos a ingresar:

 Nombre del Plan: "Sano Deleite"

 Setear al menos un alimento de los ingresados a cada día y en particular a cada horario (desayuno, almuerzo y cena). Guardar días.
 Clickear botón enviar(flecha ubicada en la esquina inferior izquierda).
 - Salida esperada: Confirmación de que el plan fue enviado.
- c) **Ingresar alimento al sistema** Pre-condición 1: Hacer click en botón ingresar alimento al sistema.
 - 1) Caso 1:
 - Entrada:
 Ingresar nombre: "Papa"
 Seleccionar tipo de alimento: Verdura
 Click en confirmar (icono tick ubicado en la parte inferior).
 - Salida esperada: Se acepta el nuevo alimento informando al usuario.
 - 2) Caso 2:
 - Entrada:
 Ingresar nombre: "Frutilla"

Seleccionar tipo de alimento: Fruta

Seleccionar foto haciendo click en el icono de la cámara de fotos (elegir cualquier imagen png disponible)

Ingresar proporción de nutrientes según corresponda

Click en confirmar (icono tick ubicado en la parte inferior).

- Salida esperada: Se acepta el nuevo alimento informando al usuario.

3) Caso 3:

- Entrada: Click en confirmar (icono tick ubicado en la parte inferior).

- Salida esperada: Se informa al usuario que no se ha podido registrar el alimento

Además de utilizar estos sets de pruebas, se realizaron pruebas exploratorias, cuyo objetivo era testear la integración del sistema. En particular, la interacción entre usuario y profesionales en ítems como las consultas, los planes de alimentación, entre otros.

Si bien estas pruebas no pretenden ser esquemáticas para que el factor improvisación juegue a favor en la identificación de errores, a continuación se agregan algunas líneas de trabajo que se considera podrían ayudar.

1. Trabajar con cantidades significativas de usuarios y profesionales. En el caso del actual proyecto, estas cantidades se estiman entre 5 y 8.
2. Requerir dos planes de alimentación por parte del mismo usuario y al mismo profesional.
3. Testear las validaciones, esto quiere decir, trabajar con sets de datos erróneos para el sistema, para verificar que no se permita el registro de los mismos.
4. Recrear la siguiente situación:
Un usuario llamado Marcelo, se encuentra bajo una dieta estricta que sólo le permite ingerir alimentos durante el almuerzo. Solicita un plan semanal a un profesional y luego comienza a registrar ingestas solo en el horario del almuerzo.
NOTA: Situación ficticia base, recrear otras similares es altamente recomendable.
5. Dejar correr al sistema durante varias horas, usarlo de vez en cuando registrando algún alimento, profesional o usuario. Luego utilizarlo y evaluar resultados.
6. Reproducir algunos de los casos de prueba planificados y luego iniciar la sesión exploratoria.

5.3. Sesiones de ejecución de pruebas

Resultó fundamental documentar los pasos llevados adelante durante cada sesión- en caso de querer replicar algún error encontrado- así como los resultados obtenidos. Para definir las

sesiones, se analizaron las principales funcionalidades de la aplicación y también la relación que pudiera existir entre ellas.

Se definió como estrategia que las sesiones que cubrían las funcionalidades de mayor prioridad fueran realizadas por ambas partes del equipo de trabajo, para así asegurar su mayor seguridad y prueba exhaustiva.

Entornos utilizados para las pruebas

1. Computadora HP modelo envy dv6
2. Computadora HP envy x360 Convertible
3. Computadora de los laboratorios de la facultad: DELL XPS
4. ASUS Zenbook

Sesiones de prueba vinculadas a funcionalidades generales del sistema

- **Sesión 1:** asociada al caso de prueba 1.A.1.

Fecha: 13/11/17

Versión: 2.0.0

Duración: 4 minutos

Entorno específico: 2

Resultado: El sistema actúa de acuerdo a lo esperado, aparecen iconos en pantalla indicando que los datos ingresados son correctos. Resulta poco práctico el calendario para seleccionar la fecha, teniendo en cuenta que si la fecha es asociada al nacimiento hay que ir una cantidad de años para atrás de manera manual. La fecha no se selecciona en todos los casos al presionar una única vez, sino que hay que hacerlo de modo iterativo.

- **Sesión 2:** asociada al caso de prueba 1.A.2.

Fecha: 13/11/17

Versión: 2.0.0

Duración: 6 minutos

Entorno específico: 2

Resultado: Las validaciones funcionan de acuerdo a lo esperado con excepción de la fecha de nacimiento que puede no seleccionarse y de todos modos es posible el registro.

- **Sesión 3:** asociada al caso de prueba 1.A.2.

Fecha: 13/11/17

Versión: 2.0.0

Duración: 3 minutos

Entorno específico: 2

Resultado: Al subir una foto de perfil ocurre un error con la ruta de la imagen. El sistema se cae.

- **Sesión 4:** asociada al caso de prueba 1.B.1.
 Fecha: 18/11/2017
 Versión: 3.2.0
 Duración: 4 minutos
 Entorno específico: 1
 Resultado: La ventana de registro de profesional tiene un tamaño mucho mayor al esperado, y se diferencia de las demás. Lo demás funciona correctamente.
- **Sesión 5:** asociada al caso de prueba 1.B.2.
 Fecha: 18/11/2017
 Versión: 3.2.0
 Duración: 6 minutos
 Entorno específico: 1
 Resultado: Las validaciones funcionan de manera correcta. Se repiten los errores de la sesión 2 con respecto al tamaño de la ventana y la sesión 1.1 con respecto a la validación de la fecha de nacimiento y graduación.
- **Sesión 6:** asociada a la serialización del sistema.
 Fecha: 10/11/2017
 Versión: 2.0.1
 Duración: 9 minutos Entorno específico: 3
 Resultado: Cierro el sistema luego de las sesiones anteriores. Al abrirlo nuevamente, se encuentran todos los datos de los usuarios y/o profesionales registrados.
- **Sesión 7:** asociada al caso de prueba 1.C.1
 Fecha: 10/11/2017
 Versión: 2.0.1
 Duración: 2 minutos
 Entorno específico: 3
 Resultado: El sistema no se cae pero tampoco notifica que no hay usuarios registrados en el mismo. Simplemente no hace nada.
- **Sesión 8:** asociada al caso de prueba 1.C.2
 Fecha: 9/11/2017
 Versión: 2.0.0
 Duración: 7 minutos
 Entorno específico: 4
 Resultado: Aparecen todas las opciones del menú de usuario de la manera esperada.
- **Sesión 9:** asociada al caso de prueba 1.D.1
 Fecha: 9/11/2017
 Versión: 2.0.0
 Duración: 11 minutos
 Entorno específico: 4
 Resultado: Ocurre de manera análoga a la Sesión número 4.
- **Sesión 10:** asociada al caso de prueba 1.D.2
 Fecha: 9/11/2017

Versión: 2.0.0

Duración: 13 minutos

Entorno específico: 4

Resultado: Aparecen todas las opciones del menú del profesional de la manera esperada.

Sesiones de prueba vinculadas a funcionalidades dentro del rol de usuario

- **Sesión 1:** asociada al caso de prueba 2.A.1

Fecha: 10/11/2017

Versión: 2.1.1

Duración: 8 minutos

Entorno específico: 2

Resultado: Salida esperada coincide con la salida resultante. Aparece un cartel en pantalla indicando que no hay profesionales aún registrados. El botón de consulta con profesional se marca como seleccionado con un tono gris distinto del negro habitual, lo que le da la sensación al usuario de donde está ubicado.

- **Sesión 2:** asociada al caso de prueba 2.A.2

Fecha: 10/11/2017

Versión: 2.1.1

Duración: 7 minutos

Entorno específico: 2

Resultado: Aparece en pantalla el cartel para crear una nueva conversación. Al presionarlo, se muestra una ventana que no es del todo armónica en cuanto a la estética, pero que funciona correctamente. Al seleccionar el profesional que se quiere contactar, se despliega otro nuevo panel en donde se puede enviar una cantidad de mensajes. En caso de querer solicitar una nueva conversación con otro profesional, el sistema no muestra ningún elemento. Esto se debe a que solo existe un único profesional registrado en el sistema.

- **Sesión 3:** asociada al caso de prueba 2.A.3

Fecha: 10/11/2017

Versión: 2.1.1

Duración: 16 minutos

Entorno específico: 2

Resultado: Se reitera muchas veces el mismo ciclo; entrar a la aplicación ingresado como determinado usuario, enviar un mensaje a determinado profesional, cerrar sesión y volver a iniciarla. Las conversaciones persisten de manera correcta a pesar del inicio y cierre de sesión. Se descubre que el sistema permite que el usuario envíe una infinidad de mensajes vacíos.

- **Sesión 4:** asociada al caso de prueba 2.A.3

Fecha: 10/11/2017

Versión: 2.1.1

Duración: 4 minutos

Entorno específico: 2

Resultado: El nombre del profesional registrado en el sistema es muy extenso. Queda cortado en pantalla.

- **Sesión 5:** asociada al caso de prueba 2.A.3

Fecha: 10/11/2017

Versión: 2.1.1

Duración: 5 minutos

Entorno específico: 2

Resultado: El profesional tenía registrada una foto de perfil, pero al ingresar a la consulta de profesionales la misma no aparece seteada al lado de su nombre.

- **Sesión 6:** asociada al caso de prueba 2.A.4

Fecha: 10/11/2017

Versión: 2.1.1

Duración: 10 minutos

Entorno específico: 2

Resultado: Se reitera que en caso de existir otros profesionales en el sistema aparece la lista. Si se diera que el usuario ya se contactó previamente con todos los profesionales registrados, el sistema no muestra nada.

- **Sesión 7:** asociada al caso de prueba 2.B.1

Fecha: 17/11/17

Versión: 3.0.0

Duración: 3 minutos

Entorno específico: 3

Resultado: No hay profesionales en el sistema. Se notifica que no es posible solicitar un plan de alimentación de manera correcta.

- **Sesión 8:** asociada al caso de prueba 2.B.2

Fecha: 17/11/17

Versión: 3.0.0

Duración: 7 minutos

Entorno específico: 3

Resultado: Al registrar un profesional para replicar las condiciones de ejecución de la prueba, se encontró que la label que tiene el título se encuentra cortado por la mitad, no es posible su legibilidad. Independientemente de eso, no se encontraron anomalías. El plan pudo ser solicitado, se editaron las restricciones y preferencias, el sistema informa de dicha edición. No se puede saber a esta altura de la ejecución si los datos fueron guardados de manera correcta en el sistema.

- **Sesión 9:** asociada al caso de prueba 2.B.3

Fecha: 19/11/17

Versión: 3.0.0

Duración: 6 minutos

Entorno específico: 3

Resultado: Resultado esperado igual al resultado obtenido.

- **Sesión 10:** asociada al caso de prueba 2.B.4
 Fecha: 19/11/17
 Versión: 3.0.0
 Duración: 17 minutos
 Entorno específico: 3
 Resultado: Ingreso a la opción plan de alimentación del menú y presiono el botón para consultar los planes pendientes de ese usuario. Como no existe ninguno, se despliega en pantalla el respectivo cartel. De manera instantánea, solicito un plan de alimentación y completo el formulario. Luego de pedir el plan, el sistema completamente trancado; no se actualiza la lista de planes pendientes del usuario al apretar el botón, ni tampoco se puede volver a solicitar un nuevo plan de alimentación. Solo resta presionar otra opción del menú.

- **Sesión 11:** asociada al caso de prueba 2.B.4
 Fecha: 19/11/17
 Versión: 3.0.0
 Duración: 20 minutos
 Entorno específico: 3
 Resultado: Luego de iterar varias veces sobre la misma funcionalidad se descubre que si el usuario solicita un plan de alimentación, siempre podrá verlo en la opción de buscar sus planes. Sin embargo, en caso de que el profesional no haya contestado aún a dicho plan, al apretar el nombre de la lista el obligatorio lanza una excepción.

- **Sesión 12:** asociada al caso de prueba 2.C.1
 Fecha: 19/11/17
 Versión: 3.0.0
 Duración: 4 minutos
 Entorno específico: 4
 Resultado: Funciona de manera esperada, incluso cuando no hay profesionales en el sistema. Un detalle es que el título no se encuentra alineado con respecto al ícono en pantalla.

- **Sesión 13:** asociada al caso de prueba 2.C.2
 Fecha: 19/11/17
 Versión: 3.0.0
 Duración: 7 minutos
 Entorno específico: 4
 Resultado: Se realiza de manera iterativa. En una primera instancia funciona de manera correcta. Tiene como defecto que al ingresar alimentos de manera seguida el calendario queda seteado con la fecha de ingesta del registro anterior.

- **Sesión 14:** asociada al caso de prueba 2.C.3
 Fecha: 21/11/2017
 Versión: 3.5.0
 Duración: 4 minutos
 Entorno específico: 4

Resultado: Se valida de manera exitosa la selección de un alimento. Se deja de lado las validaciones correspondientes al calendario.

Sesiones de prueba vinculadas a funcionalidades generales del profesional

- **Sesión 1:** asociada al caso de prueba 3.A.1

Fecha: 21/11/2017

Versión: 3.5.0

Duración: 2 minutos

Entorno específico: 4

Resultado: igual al esperado, el sistema avisa que no hay consultas pendientes para ese profesional.

- **Sesión 2:** asociada al caso de prueba 3.A.2

Fecha: 21/11/2017

Versión: 3.5.0

Duración: 8 minutos

Entorno específico: 4

Resultado: El profesional puede ver de manera correcta el mensaje enviado por el usuario. Sin embargo, al repetir el caso de prueba de manera reiterada se puede notar que al acceder al perfil de usuario si bien se muestran de manera correcta sus restricciones y preferencias, las ingestas no aparecen aunque existan en el sistema.

- **Sesión 3:** asociada al caso de prueba 3.B.1

Fecha: 21/11/2017

Versión: 3.5.0

Duración: 4 minutos

Entorno específico: 3

Resultado: El sistema muestra todas las veces que no hay planes de alimentación pendientes, aún cuando hay una decena de profesionales registrados.

- **Sesión 4:** asociada al caso de prueba 3.B.2

Fecha: 22/11/2017

Versión: 3.5.0

Duración: 10 minutos

Entorno específico: 3

Resultado: La lista con los usuarios que solicitaron planes de alimentación aparece de manera correcta. En caso de editar las restricciones y/o preferencias a la hora de solicitar el plan de alimentación, el profesional puede ver esas ediciones a la hora de mostrar el perfil de usuario. La foto de perfil del usuario se muestra bien, pero existe un problema con la fecha de nacimiento, que se etiqueta como no ingresada.

- **Sesión 5:** asociada al caso de prueba 3.B.3

Fecha: 22/11/2017

Versión: 3.5.0

Duración: 22 minutos

Entorno específico: 3

Resultado: Se recrea la situación para llegar al panel en donde el profesional debe de completar el plan de alimentación. En caso de que el individuo se olvide de guardar el día correspondiente, los datos se pierden. Además, hay un intento de validar el nombre del plan de alimentación mediante una label y un ícono, pero en caso de presionar el botón de continuar con el campo del nombre vacío, el plan es aceptado de todos modos.

- **Sesión 6:** asociada al caso de prueba 3.B.3

Fecha: 20/11/2017

Versión: 3.5.0

Duración: 11 minutos

Entorno específico: 1

Resultado: Al guardar los datos en la elaboración del plan de alimentación por parte del profesional, el sistema debería de indicar de algún modo que los datos fueron guardados, pero no muestra elementos. El mecanismo de agregar/eliminar distintos alimentos dado los diferentes días y comidas funciona de manera correcta. Sin embargo, el no haber encontrado defectos en estas sesiones no puede afirmar su inexistencia, al tratarse de una ventana con una gran complejidad e infinitos casos límites.

- **Sesión 7:** asociada al caso de prueba 3.C.1

Fecha: 20/11/2017

Versión: 3.5.1

Duración: 8 minutos

Entorno específico: 1

Resultado: Se registra el alimento y se notifica de manera correcta. Sin embargo, en la ventana que muestra el mensaje de registro exitoso quedaron algunos íconos y labels pertenecientes a la ventana anterior de registro que no fueron ocultados. Aún más, se trasladó de una ventana a otra el ícono que confirma el registro de un alimento, y es posible presionarlo, lo cual tranca la línea de ejecución del sistema.

- **Sesión 8:** asociada al caso de prueba 3.C.2

Fecha: 22/11/2017

Versión: 3.6.0

Duración: 6 minutos

Entorno específico: 1

Resultado: Al ingresar alimentos de manera reiterada los checkbox asociados a los nutrientes permanecen seleccionados de un ingreso a otro. Lo mismo sucede con los JTextField en caso de haber ingresado alguna proporción. Las validaciones funcionan de manera esperada; no se permite el ingreso de números negativos o letras. Falta que se limpien los campos de un registro a otro.

- **Sesión 9:** asociada al caso de prueba 3.C.3

Fecha: 22/11/2017

Versión: 3.6.0

Duración: 6 minutos

Entorno específico: 1

Resultado: Las validaciones funcionan bien, no es posible registrar alimentos dejando

el campo de nombre o tipo de alimento vacío. Incluso al cambiar el ítem de la lista el ícono cambia de manera adecuada.

6. Reporte de defectos

6.1. Definición de categorías de defectos

La información más relevante surgida a partir de las pruebas funcionales fue el reporte de defectos. Para su mayor análisis se organizaron en diferentes categorías de acuerdo a sus características. Las mismas fueron:

- **Defectos que alteran el flujo normal de utilización:** aquellas fallas y/o errores que provocan un desconcierto total por parte del usuario, e impiden la continuación del curso normal de ejecución.
- **Posibles mejoras en relación a la usabilidad del sistema:** aluden a todos los cambios que podrían ser implementados con el objetivo de mejorar el atributo de calidad; aumentar la eficiencia y eficacia.
- **Posibles mejoras en relación a la interfaz o diseño del sistema:** se conforman de transformaciones que podrían aportar en relación a la estética de la aplicación. Generalmente los defectos que comprenden esta categoría no pudieron ser mejorados como consecuencia de los altos conocimientos de Netbeans necesarios.
- **Otro tipo de errores:** elementos que no son abarcados por ninguna de las categorías anteriores, pero que existen en el sistema.

A su vez, cada categoría se compone de tres atributos que indican la severidad del defecto hallado:

- **Alta:** bugs que alteran procesos fundamentales del sistema, o incluso detienen su ejecución. El sistema no cumple con uno o más requerimientos especificados.
- **Media:** los requerido por el sistema se cumple de manera exitosa, pero quizás podría implementarse de un modo más sencillo. Incluye problemas en la presentación de la información, falta de alguna validación o mostrar en pantalla elementos no deseados.
- **Baja:** no afectan el flujo principal o alternativo del sistema, están orientados a la navegabilidad y diseño. Posición de los objetos en pantalla, alineación de los campos y mala utilización de los colores son algunos ejemplos.

Por último, luego de cada defecto presentado se indica si fue posible su corrección o no, junto con una breve explicación de los motivos que acompañan dicha decisión.

6.2. Defectos encontrados por iteración

Defectos que alteran el flujo normal de utilización

Subir al sistema una foto de perfil de un Usuario, Profesional o Alimento.

- Descripción: A la hora de seleccionar y cargar en el sistema una foto representativa- en cualquiera de las opciones de registro de los items nombrados anteriormente- puede ocurrir que en caso de que la imagen exceda determinado tamaño, la misma quede cortada. Este defecto no se puede visualizar únicamente en las ventas de registro, sino que a posterior, en todas las ventanas que se muestra la imagen se verá de ese modo.
- Severidad: Alta
- Corrección: No realizada

Registrar en el sistema dos alimentos o dos ingestas de manera consecutiva.

- Descripción: Al ingresar un alimento por parte del profesional, se requiere llenar un formulario con determinados campos. Luego de realizada la tarea de manera exitosa, se le notifica a través de un nuevo panel en pantalla. En caso de decidir volver a ingresar otro alimento de manera continua, al abrirse el formulario de registro se encontrarán setedas las labels indicatorias de la correctitud o error de los datos- correspondientes al primer alimento ingresado. Lo mismo ocurre con las proporciones de los nutrientes para el alimento.
- Severidad: Alta
- Corrección: Realizada. Fue considerado vital para la buena navegación en el sistema y la eficiencia de quien lo utiliza.

Defecto en el modo en que se ve la pantalla al cambiar de dispositivo o de resolución.

- Descripción: En determinado momento del desarrollo del proyecto, fue preciso percatarse que el proyecto de Netbeans no se comportaba del mismo modo al cambiar su equipo de ejecución. En ocasiones ocurría que la pantalla aumentaba o disminuía su tamaño de acuerdo a la resolución de pantalla predeterminada, o ciertas ventanas no se veían de la manera esperada. El defecto resulta crítico en tanto desencadenaba en que no fuera posible acceder a botones de carácter prioritario dentro de la pantalla. Por ejemplo, en la computadora HP ENVY dv6 con resolución 1360x768 la pantalla no puede verse en su totalidad, cortando así el acceso por parte del usuario a funcionalidades.
- Severidad: Alta

- Corrección: Se decidió tomar como estándar la resolución y tamaño de pantalla establecidos en los laboratorios de la facultad (1360x768). Si bien el problema no se resolvió en su totalidad, se recurrió a un modelo de referencia para la elaboración de la interfaz.

Esperar respuesta a un plan de alimentación por parte del Usuario.

- Descripción: Luego de que el usuario solicita un plan de alimentación, debe de esperar a que el mismo sea completado por parte del profesional. El usuario puede realizar una búsqueda de sus planes de alimentación en el sistema. Si el plan solicitado anteriormente fuera el único por parte del usuario, a la hora de realizar esa búsqueda aparecería en pantalla una lista anunciando que aún “no hay planes de alimentación recibidos”. Dicha lista es de carácter JList, y al presionarla, el sistema se cae lanzando una excepción.
- Severidad: Alta
- Corrección: Realizada. Fue tomado como prioridad en tanto no es aceptable una caída del sistema en ningún momento de su ejecución.

Registro de alimentos por parte de un profesional.

- Descripción: En la ventana correspondiente al registro de alimentos por parte de un profesional es posible seleccionar de un conjunto de checkbox los nutrientes que pertenezcan al nuevo alimento que está siendo creado. En caso de seleccionar determinado nutriente, emergerá a su derecha un JTextField para escribir la proporción en gramos que acompaña al nutriente. Si sucediera que un profesional selecciona determinado checkbox, deja el JTextField en blanco y presiona el botón de crear un nuevo alimento, el sistema se cae.
- Severidad: Alta
- Corrección: Realizada. Se valida que en caso de que la proporción del nutriente seleccionado quede vacía, se le setea una por defecto del sistema.

Menú principal al ingresar al sistema como Profesional o Usuario.

- Descripción: Al seleccionar cualquiera de los dos tipos de usuario emerge una ventana que contiene el menú de opciones del lado izquierdo, y un panel negro (vacío) a su derecha. La ventana de inicio es la primera impresión que el individuo tiene de la aplicación, y no es aceptable que la misma se muestre vacía. Debería contener elementos que muestren información relevante; datos estadísticos sobre su alimentación, ingestas sugeridas, profesionales para contactar, eventos relacionados con nutrición cercanos a su ubicación, entre otros.
- Severidad: Alta
- Corrección: No realizada por motivos asociados al tiempo de entrega.

Librería JCalendar no encontrada al abrir el proyecto en Netbeans.

- Descripción: Para mejor manejo de las fechas dentro de la aplicación, fueron utilizados los calendarios provistos por la librería JCalendar de Java. Sin embargo, cada vez que se vuelve a abrir el proyecto en Netbeans, la ruta a dicha librería se pierde, y hay que redestinarlo a mano para que el proyecto compile.
- Severidad: Alta
- Corrección: Realizada. Resultaba incómodo tener que buscar entre todas las carpetas cada vez que se volvía a cargar el proyecto. Aún más, otra persona que no integrara el equipo de trabajo de este proyecto podría no tener los conocimientos sobre donde se encuentra la librería en las carpetas del proyecto.

Extensión de las imágenes a la hora de subir una foto de perfil o de un alimento.

- Descripción: Fue encontrado un error a la hora de seleccionar una foto de la computadora para subir al sistema. La extensión de los archivos resultaba equivocada por un error de tipografía en el método, lo que dificultaba su búsqueda.
- Severidad: Baja
- Corrección: Realizada. Mejora sencilla. Se destaca que en el proceso de mejora del defecto se realizó una sesión de prueba en donde el sistema se caía como producto de permitir la selección de archivos que no fueran imágenes, lo cual fue corregido.

Error al visualizar el perfil de un determinado usuario.

- Descripción: Cuando un profesional responde a una consulta por parte de un usuario, tiene como funcionalidad la opción de poder ver el perfil de dicho usuario para evaluar la consulta de acuerdo a sus preferencias o restricciones. No obstante, el usuario no tiene por qué completar los campos anteriores en su registro. Si las listas que contienen las preferencias o restricciones se encuentran vacías, el sistema se cae al acceder a la ventana de perfil de usuario.
- Severidad: Alta
- Corrección: Realizada

Error al visualizar la lista de ingestas en el perfil de usuario.

- Descripción: Luego de solucionar el defecto anterior se agrega una nueva sesión de prueba en donde se descubre que si bien se solucionó de manera parcial la visualización del perfil de usuario en cuanto a las restricciones y preferencias, la lista de ingestas continúa sin aparecer.

- Severidad: Alta
- Corrección: Realizada

Solicitud de plan de alimentación por parte del usuario en caso de no haber profesionales registrados en el sistema.

- Descripción: El usuario presiona el icono correspondiente a la solicitud de un plan de alimentación- sin haber profesionales registrados en el sistema. La ventana se abre de todos modos, se despliega una lista de profesionales vacía.
- Severidad: Alta
- Corrección: Realizada. Consistió simplemente en poner una validación y un nuevo panel.

Solicitud de más de un único plan de alimentación entre un mismo Profesional y un Usuario.

- Descripción: No existen problemas en el ciclo relacionado a solicitar-completar-visualizar un plan de alimentación entre ambas partes. Sin embargo, llegado el caso en que el Usuario quiere solicitar nuevamente un plan de alimentación con el mismo Profesional, si bien puede hacerlo, al Profesional no se le muestra una instancia para poder completarlo.
- Severidad: Alta
- Corrección: No realizada por falta de tiempo.

La fecha máxima que el calendario permite seleccionar no coincide con la fecha del día actual.

- Descripción: La fecha máxima que es posible seleccionar dentro del calendario no se actualiza de acuerdo al paso de los días, sino que quedo establecida como predeterminada la del día en que se posicionó el calendario en la ventana.
- Severidad: Alta
- Corrección: Solucionado a través de la clase `GregorianCalendar` y el método `.setMaxDate`

Posibles mejoras en relación a la usabilidad del sistema

El programa no está desarrollado sobre una única ventana de interfaz

- Descripción: Existen casos que surgen ventanas emergentes, lo cual no es lo ideal. Ejemplos de lo anterior son; registro de usuario o profesional, editar las preferencias por parte de un usuario, ver el perfil de usuario, entre otros.

- Severidad: Baja.
- Corrección: El defecto no fue solucionado. Se consideraron otros elementos a mejorar como prioritarios en tanto las ventanas no interrumpen la fluidez del programa.

Elaborar plan de alimentación por parte de un Profesional.

- Descripción: La ventana para completar el plan de alimentación solicitado contiene algunos defectos; el Profesional no puede ir visualizado cuales son los días que va completando, e incluso en caso de olvidarse de presionar el botón de guardar se pierden todos los datos completados. Quizás se podría implementar la ventana de un modo mas sencillo desde el punto de vista de la lógica para facilitar su uso. Además, tampoco se le notifica que los elementos fueron guardados en ese día.
- Severidad: Media
- Corrección: No realizada

Ingreso de campo de fecha de una determinada ingesta

- Descripción: En el registro de ingesta, se le pide al usuario que seleccione manualmente la fecha de la misma de un JCalendar. Una mejora podría consistir en tomar la fecha directamente de la computadora, ya que se asume que si se está ingresando una ingesta, la misma ocurrió ese mismo día.
- Severidad: Media
- Corrección: No realizada.

Campos no alineados en determinados formularios de registro

- Descripción: La organización de ciertas ventanas dentro del sistema no permitieron el espacio suficiente como para que todos los campos estuvieran perfectamente alineados.
- Severidad: Media
- Corrección: No realizada.

La fecha ingresada en un registro se mantiene de una ventana a otra

- Descripción: En caso de ingresar muchos profesionales, usuarios o ingestas de manera reiterada, se mantiene de una ventana a otra la fecha seleccionada anteriormente.
- Severidad: Baja
- Corrección: No realizada. No altera al flujo de ejecución.

Selección de usuarios en la ventana de inicio del sistema

- Descripción: Al comenzar a ejecutarse la aplicación, se muestran dos listas con los usuarios y profesionales registrados hasta el momento, junto con una opción para agregar nuevos individuos de ambos tipos. No se requiere ningún tipo de autenticación, lo cual está vinculado a las restricciones del obligatorio. Sin embargo, esta metodología podría resultar poco practica en caso de tratarse de un sistema de una cantidad masiva de usuarios.
- Severidad: Baja
- Corrección: No realizada. Se supone un número de usuarios pequeño y resulta una forma rápida de entrar al sistema.

Cumplimiento de la heurística de ayuda y documentación

- Descripción: La aplicación no provee ningún tutorial que le indique al usuario cuales son las opciones básicas y funcionalidades del sistema.
- Severidad: Alta
- Corrección: Realizada

Ventana emergente a la hora de elegir un profesional para iniciar una consulta por parte del usuario.

- Descripción: Dentro de las funcionalidades del usuario, se encuentra la de consulta con profesionales. En caso de querer enviarle un mensaje a un profesional que aún no tiene conversaciones, se cambia el panel en pantalla por uno que contiene únicamente la lista de profesionales en el sistema con el objetivo de que seleccione. Quizás la posible mejora podría incluir que no se utilice una pantalla específicamente para mostrar la lista anterior, sino que aparezca al costado, de manera simultánea al resto de los elementos en pantalla.
- Severidad: Media
- Corrección: No realizada

No se muestra el status del sistema en consulta de profesional en caso de no existir profesionales registrados.

- Descripción: Si un usuario deseara enviarle un mensaje a profesional y presiona el respectivo icono, en caso tal de no haber profesionales registrados en el sistema, simplemente no sucede nada. Lo mismo ocurre si el usuario ha contactado a todos los profesionales existentes. El sistema debería dar un aviso al usuario de lo que está ocurriendo.

- Severidad: Media
- Corrección: Relativa. Se agregó un panel indicador de la inexistencia de profesionales registrados, pero no se resolvió el caso en que el usuario haya contactado a todos los profesionales registrados.

El usuario no tiene la posibilidad de ver sus ingestas anteriores al realizar una consulta con profesional.

- Descripción: Una de las posibles funcionalidades de la consulta con profesionales es enviar mensajes sobre las propiedades de los diferentes alimentos ingeridos. Sería de mayor utilidad para el usuario que consulta tener una opción donde pudiese ver las ingestas ingresadas en el último tiempo.
- Severidad: Media
- Corrección: No realizada.

Posibles mejoras en relación a la interfaz o diseño del sistema

Falta de visibilidad de algunas nacionalidades en el registro de usuario

- Descripción: Algunos nombres de países quedaban cortados en pantalla a la hora de desplegar el Combo Box.
- Severidad: Baja
- Corrección: Realizada

Listas en pantalla de distintos colores a lo largo de la navegación en la aplicación.

- Descripción: Las listas que contienen a los usuarios y profesionales en el Inicio de Sesión son de color negro con letras blancas, mientras que la lista para seleccionar un profesional y crear una conversación es blanca con letras negras. No se mantienen los estándares de estilo.
- Severidad: Baja
- Corrección: No realizada.

Listas mostradas en pantalla en el inicio de sesión en caso de no haber registrado usuarios o profesionales.

- Descripción: En caso de ejecutar el sistema sin datos en su serialización, es decir sin profesionales o usuarios registrados, se mostrará la ventana de inicio de sesión con dos listas vacías, lo cual no es deseable desde el punto de vista de la estética.

- Severidad: Baja
- Corrección: No realizada.

Ventana emergente File Chooser provista por la librería de Java

- Descripción: A la hora de subir una foto al sistema, al presionar el ícono descriptivo con una cámara, se abre un File Chooser con el objetivo de que el individuo seleccione de su computadora la imagen correspondiente. Este sistema de archivos, no cumple con la consistencia y estilos que la aplicación intenta mantener a lo largo de toda su navegación. Aún más, la legibilidad en el mismo no es del todo clara debido al contraste entre los colores.
- Severidad: Media
- Corrección: No realizada.

Existen ventanas que no cuidan el estilo minimalista de la aplicación.

- Descripción: Pantallas como la de elaborar un plan de alimentación por parte de un profesional, o la de ver uno realizado por el usuario, contienen paneles que pueden considerarse como sobrecargadas de información con respecto a las demás de la aplicación.
- Severidad: Media
- Corrección: No realizada por falta de tiempo. Se requiere rediseñar y volver a organizar toda la interfaz.

Mala visibilidad y selección del JCalendar en las opciones de registro.

- Descripción: El calendario provisto por la librería de Java no resulta muy cómodo para quien lo utiliza a la hora de seleccionar cierta fecha. Hay que presionar más de una vez y a veces no muestra lo que el usuario pretende.
- Severidad: Baja
- Corrección: No realizada

Posibilidad de seleccionar de la lista de conversaciones al no haber creado alguna aún.

- Descripción: En la consulta de profesional por parte del usuario, al no haber conversaciones creadas aparece una lista que lo notifica. Sin embargo, es posible seleccionar items de esta lista, tales como si fueran profesionales, lo cual no es deseado en cuando a la estética del proyecto. Se podría solucionar agregando un panel.
- Severidad: Baja

- Corrección: Realizada.

Otro tipo de errores

Falta de validación de la fecha de nacimiento o fecha de graduación dentro del JCalendar.

- Descripción: No fue posible validar el caso en que un usuario o profesional no seleccione ninguna fecha de nacimiento o graduación dentro del JCalendar. El sistema tomará como predeterminada la fecha del día de hoy, en vez de notificarle al usuario que resulta necesario insertarla.
- Severidad: Alta
- Corrección: No realizada. Se requiere una investigación en profundidad sobre la librería.

Solo es posible subir al sistema imágenes con extensión png

- Descripción: Con el objetivo de validar que el usuario no pueda subir como foto de perfil cualquier archivo de su computadora, se seleccionó un tipo predeterminado de extensión, lo que excluye a otras imágenes válidas.
- Severidad: Baja
- Corrección: No realizada.

Problemas de vista de página anterior

- Descripción: Luego de registrar una ingesta o un alimento en el menú principal del usuario y del profesional respectivamente, aparece un cartel de confirmación, lo cual sigue con el curso normal de ejecución. Sin embargo, al deslizar el mouse sobre el panel, se hacen visible elementos del panel anterior, esto quiere decir que en el caso del registro de ingesta, se muestra campos de dicha ventana aunque ya hayan sido completados. Ídem para registro de alimentos.
- Severidad: Media
- Corrección: Realizada. Faltaba ocultar el panel anterior.

6.3. Estado de calidad global

El primer aspecto a destacar es que se logró abarcar todas las funcionalidades incluidas en la rúbrica. Se logró superar obstáculos tales como el desconocimiento de algunas herramientas necesarias para desarrollar el sistema, lo cual quedo en manos del equipo para investigar.

La primera conclusión es que hubiera sido completamente imposible cumplir con un proyecto de tal magnitud sin encontrar a posterior una gran cantidad de defectos. Inclusive, sistemas que utilizamos de manera diaria los tienen, es inherente al desarrollo del software.

Resultaron exitosas las estrategias de testing utilizadas- guiadas por una buena práctica de sesiones de prueba. Analizar las posibles entradas del sistema permitió hacer énfasis en encontrar errores en las funcionalidades más críticas. De hecho, muchos de los errores encontrados resultaron vitales de corregir, para asegurar el correcto funcionamiento de la aplicación y por ende fueron debidamente atendidos.

Sin embargo, otros defectos no pudieron ser arreglados por falta de tiempo o de conocimiento en el equipo de trabajo. El clasificar y categorizar los errores permitió priorizar la corrección de los que entraron en la categoría ALTA.

Algunos defectos son en realidad mejoras en cuanto a usabilidad y diseño que podrían ser implementadas con un mayor margen de tiempo y recursos.

En general, el código presenta una buena consistencia y coherencia que permite un uso sin mayores problemas de las funcionalidades, todo esto, está avalado por un alto nivel de testing unitario y un posterior testing funcional, que intentó maximizar los resultados obtenidos.