**Lybel**

**Plan de Verificación y Validación**

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 22/05/2020 | 1.0 | Versión Inicial | Fernando Figuereo |
| 30/05/2020 | 1.1 | Modificaciones al plan | Fernando Figuereo |
| 10/06/2020 | 1.2 | Revisión | Wilson Gómez |
| 10/06/2020 | 1.2 | Versión estable | Wilson Gómez |
| 12/06/2020 | 1.3 | Modificaciones al plan | Wilson Gómez |
| 16/06/2020 | 1.4 | Correcciones mínimas | Fernando Figuereo |
| 16/06/2020 | 1.5 | Revisión | Wilson Gómez |
| 07/06/2020 | 1.6 | Modificación al plan, prueba de performance | Fernando Figuereo |
| 08/06/2020 | 1.7 | Pruebas Unitarias | Fernando Figuereo |
| 09/06/2020 | 1.8 | Revisión | Wilson Gómez |
| 09/06/2020 | 1.9 | Modificación al plan | Wilson Gómez |
| 09/06/2020 | 2.0 | Versión estable | Wilson Gómez |
| 12/06/2020 | 2.0 | Revisión | Fernando Figuereo |
| 15/06/2020 | 2.1 | Versión Final | Fernando Figuereo |

**Contenido**

[**1.** **Introducción**](about:blank) **3**

[1.1. Propósito](about:blank) 3

[1.2. Punto de partida](about:blank) 3

[1.3. Alcance](about:blank) 3

[1.4. Identificación del proyecto](about:blank) 3

[1.5. Estrategia de evolución del Plan](about:blank) 3

[**2.** **Requerimientos para verificar**](about:blank) **3**

[**3.** **Estrategia de Verificación**](about:blank) **4**

[3.1. Tipos de pruebas](about:blank) 4

[*3.1.1.* *Prueba de integridad de los datos y la base de datos*](about:blank) *4*

[*3.1.2.* *Prueba de Funcionalidad*](about:blank) *4*

[*3.1.4.* *Prueba de Interfase de Usuario*](about:blank) *5*

[*3.1.5.* *Prueba de Performance*](about:blank) *6*

*pruebas unitarias*

[*3.1.9.* *Prueba de Seguridad y Control de Acceso*](about:blank) *9*

[*3.1.10.* *Prueba de Fallas y Recuperación*](about:blank) *9*

[*3.1.11.* *Prueba de Configuración*](about:blank) *11*

[*3.1.12.* *Prueba de Instalación*](about:blank) *11*

[*3.1.13.* *Prueba de Documentos*](about:blank) *12*

[3.2. Herramientas](about:blank) 13

[**4.** **Recursos**](about:blank) **13**

[4.1. Roles](about:blank) 13

[4.2. Sistema](about:blank) 14

[**5.** **Hitos del proyecto de Verificación**](about:blank) **14**

[**6.** **Entregables**](about:blank) **14**

[6.1. Modelo de Casos de Prueba](about:blank) 15

[6.2. Informes de Verificación](about:blank) 15

[6.3. Evaluación de la verificación](about:blank) 16

[6.4. Informe final de verificación](about:blank) 16

**7**[**.** **Apéndice**](about:blank) **19**

[9.1. Niveles de gravedad de error](about:blank) 19

[9.2. Niveles de aceptación para lo elementos verificados](about:blank) 19

**Introducción**

* 1. **Propósito**

Entre los propósitos de este proyecto están:

* La verificación del estado actual del software y los componentes en él.
* Definir las pautas que seguiremos para su reparación.
* Definir los recursos necesarios para el proyecto.
* Enumerar los entregables.
  1. **Punto de partida**

Se verificará el sistema Ventas Lybel. Este consiste en una aplicación de Punto de Ventas para el seguimiento y control de los productos de una empresa. Esta app permitirá a los usuarios registrarse, realizar búsquedas, modificar, eliminar, agregar productos así también realizar vistas de previa de los productos.

* 1. **Alcance**

En el avance de nuestro proyecto se irán realizando pruebas con el framework de JUNIT para ir confirmando los avances en el software. Las pruebas de esfuerzo no serán realizadas ya que no contamos con las herramientas ni la información total del ciclo de negocio al el cual estará dedicado este software. Las pruebas que se realizaran en el software serán dedicadas a los requerimientos ya predefinidos en el backlog que fue entregado con el programa, probando su eficiencia y funcionalidad.

Se modificaran secciones de código en la parte visual para su correcta usabilidad entre las cuales están:

1. Cambio de las imágenes y background ya establecidos para una mejor visualización de los botones en la parte inferior en la pantalla de Login y Registro.
2. Se modificara el texto de Portada dependiendo de la vista en la que se encuentre para una mejor ubicación en el software.

Entre las modificaciones de código están:

1. Se agregaran funcionalidades de limpieza al momento de hacer una búsqueda ya que si se hace continuamente se repiten los resultados.
2. Se añadirá la función de limpieza de campos al momento de agregar un nuevo producto.
   1. **Identificación del proyecto**

Los recursos utilizados para el plan y verificación del proyecto serán los siguientes:

* Backlog de requerimientos entregados junto con el software.
  1. **Estrategia de evolución del Plan**
* Nuestro equipo (Venta Lybel) se encargará de supervisar el plan de ejecución descrito.
* Se realizarán modificaciones del presente documento según los cambios que vayan requiriendo este proyecto.
* Los cambios serán analizados por nuestro equipo de trabajo para confirmar si será necesario aplicarlo o irnos por otros cambios.
* Se realizarán nuevas documentaciones con los cambios que se estarán implementando en el sistema lo cual serán presentados al cliente y al administrador del proyecto.

**Historias a verificar:**

**Gestión de usuarios:**

1. Registro mediante usuario y contraseña.
2. Login
3. Gestión de productos
4. Añadir nuevo productos
5. Editar productos
6. Eliminar productos
7. Vista de impresión de productos
8. Buscar productos por código, nombre, precio, categoría y marca.
9. Opción de ayuda
10. **Requerimientos para verificar**

Se utilizara el framework ya mencionado JUNIT para realizar pruebas unitarias en todas las funcionalidades presentes en el software actual, y que funcionen correctamente y cumpliendo los requerimientos mencionados en el backlog.

Los requerimientos que serán verificados son los siguientes:

* Login de los usuarios
* Registro de nuevos usuarios
* Busqueda de productos
* Agregar nuevos productos
* Modificar productos ya existentes
* Eliminar productos
* Pruebas de usabilidad
* Ayuda del programa
* Acerca del programa
* Cierre de sesión
* Salida del programa

1. **Estrategia de Verificación**
   1. **Tipos de pruebas**
      1. **Prueba de integridad de los datos y la base de datos**
         1. ***Objetivo de la prueba***

Asegurar que los métodos y procesos de acceso a la base de datos funcionan correctamente y sin corromper datos

* + - 1. ***Técnica***

Se probara cada método y proceso individualmente que se utilicen hacia la base de datos, Luego se verificara la base de datos para confirmar su correcta funcionalidad y que los eventos probados funcionaron correctamente.

* + - 1. ***Criterio de aceptación***

Todos los métodos y procesos de acceso a la base de datos funcionan como fueron diseñados y sin datos corruptos.

* + - 1. ***Consideraciones especiales***
* La prueba requiere un entorno de administración de DBMS o controladores para ingresar o modificar información directamente en la base de datos, en el cual utilizamos la versión 12 de PostgreSQL y la versión 4 de pgAdmin para utilizarla.
* Los procesos deben ser invocados manualmente.
  + 1. **Prueba de Funcionalidad**
       1. ***Objetivo de la prueba***

Asegurar el correcto funcionamiento de cada una de las historias o funcionalidades del sistema.

* + - 1. ***Técnica***

Ejecute cada función usando datos válidos y no válidos, para verificar lo siguiente:

* Se obtienen los resultados esperados cuando se usan datos válidos.
* Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados, manteniendo el correcto funcionamiento del sistema.
* Se aplica apropiadamente cada regla del negocio.
  + - 1. ***Criterio de aceptación***
      2. La prueba de funcionalidad se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio. Los objetivos de estas pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio.
      3. ***Consideraciones especiales***

Ninguna hasta el momento

* + 1. **Prueba de Interface de Usuario**
       1. ***Objetivo de la prueba***

Verificar que la interfaz final provea al usuario el acceso y navegación cómoda y apropiada. Buscando cumplir los estándares de la empresa establecidos para su correcto funcionamiento.

* + - 1. ***Técnica***
* Se realizara la prueba heurística de funcionalidad y test de usabilidad para cualquier usuario.

***3.1.3.3 Criterio de aceptación***

Cada ventana ha sido verificada exitosamente siendo consistente con una versión de referencia o estándar establecido

***3.1.3.4 Consideraciones especiales***

Algunas vistas como la de inicio no estarán totalmente rellenadas ya que no se suplen requerimientos hacia esa vista.

* + 1. **Prueba de Performance**

En esta prueba se miden y evalúan los tiempos de respuesta, los tiempos de transacción y otros requerimientos sensitivos al tiempo. El objetivo de la prueba es verificar que se logren los requerimientos de performance.

***Objetivo de la prueba***

Verificar la performance de determinadas transacciones o funciones de negocio bajo ciertas condiciones:

* Condiciones de trabajo normales conocidas.
* Peores casos de condiciones de trabajo conocidas.
  + - 1. ***Técnica***
* Usar procedimientos de prueba desarrollados para verificar funciones o ciclos de negocio.
* Modificar archivos de datos para aumentar el número de transacciones o los procedimientos de prueba para aumentar el número de iteraciones de ocurrencia de transacciones.
  + - 1. ***Criterio de aceptación***

|  |  |  |
| --- | --- | --- |
| ***FUNCIONALIDAD*** | ***TIEMPO DE RESPUESTA*** | ***SE COMPLETO EN EL TIEMPO ESTIMADO***  ***(1.5 Seg)***  ***(FALLÓ/PASÓ)*** |
| Registro | ***0.6 Segs*** | ***PASÓ*** |
| Login | ***0.8 Segs*** | ***PASÓ*** |
| Añadir productos | ***0.4 Segs*** | ***PASÓ*** |
| Editar productos | ***0.5 Segs*** | ***PASÓ*** |
| Eliminar productos | ***0.2 Segs*** | ***PASÓ*** |
| Buscar productos | ***0.6* *Segs*** | ***PASÓ*** |
| Limpiar | ***0.4 Segs*** | ***PASÓ*** |
| Salir del sistema | ***0.8 Segs*** | ***PASÓ*** |

* + - 1. ***Consideraciones especiales***

N/A

* + 1. **Pruebas** 
       1. ***Objetivo de la prueba***

Poner a prueba los casos de uso del sistema, teniendo en mente lo que recibe cada uno y la respuesta que debería retornar, esto para mantener el correcto funcionamiento del sistema a medida que se agregan más características al sistema.

* + - 1. ***Técnica***

Se utilizó la librería JUNIT para llevar a cabo todas las pruebas de testing de las diferentes funcionalidades del sistema. Aquí las funcionalidades que implementan unit testing:

* Producto: Añadir, modificar, eliminar, visualizar.
* Usuario: Login, Registro
  + - 1. ***Criterio de aceptación***

Todos los test unitarios siguen pasando correctamente después de cualquier cambio agregado al sistema o implementado

* + - 1. ***Consideraciones especiales***

N/A

* + 1. **Prueba de Seguridad y Control de Acceso**
       1. ***Objetivo de la prueba***
* Seguridad en el ámbito de aplicación: Verificar que un usuario pueda acceder solo a las funciones o datos para los cuales su tipo de usuario tiene permiso.
* Seguridad en el ámbito del sistema:
* Verificar que solo los usuarios con acceso al sistema y a las aplicaciones, puedan acceder a ellos. Verificar en concreto, que sólo los usuarios registrados en el sistema tengan acceso a las funcionalidades que el mismo ofrece.
* Se comprobará que efectivamente no hay datos guardados en formato de texto plano en la base de datos, y que estarán encriptados.
  + - 1. ***Técnica***
* Seguridad en el ámbito de aplicación:
  + Identificar y hacer una lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso.
  + Crear pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario.
  + Modificar el tipo de usuario y volver a ejecutar las pruebas para los mismos usuarios. En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o son denegados.
* Acceso en el ámbito del sistema:
* Ver consideraciones especiales más abajo.
  + - 1. ***Criterio de aceptación***

Para cada tipo de actor conocido las funciones y datos apropiados están disponibles, y todas las operaciones funcionan como se espera y ejecutan las pruebas de Funcionalidad de la aplicación

* + - 1. ***Consideraciones especiales***

La persona debe registrarse a través de la aplicación para poder utilizar sus funciones de acceso a datos.

* + 1. **Prueba de Fallas y Recuperación**
       1. **Objetivo de la prueba**

Verificar que los procesos de recuperación (manual o automáticos) recuperen apropiadamente la base de datos, aplicaciones y sistema a un estado conocido y deseado.

* + - 1. **Técnica**

Se utilizaran pruebas para probar la Funcionalidad y ciclos de recuperación enfrente a situaciones que pueden ocurrir mediante el uso del software:

* Interrumpir la energía del cliente: apagar el PC.
* Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
* Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red o router
* Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de recuperación.
* Las pruebas de ciclos incompletos utilizan la misma técnica excepto que los procesos de bases de datos deben ser abortados a sí mismos o terminados prematuramente.
  + - 1. **Criterio de aceptación**

El sistema y la base de datos deben poder realizar todos los procesos de recuperación y volver a un estado conocido y estable, en el cual no haya ocurrido ninguna corrupción de datos en la base de datos y que ninguno de los requerimientos del sistema este dañado.

* + - 1. **Consideraciones especiales**

Los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como software de diagnóstico. Se requieren los grupos de recursos de Sistemas, Bases de datos y Red.

Estas pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada

* + 1. **Prueba de Configuración**
       1. ***Objetivo de la prueba***

Verificar que el software funcione apropiadamente en las configuraciones requeridas de hardware y software.

* + - 1. ***Técnica***

Verificar que el software funcione apropiadamente en las configuraciones requeridas de hardware y software.

* + - 1. ***Criterio de aceptación***

Todas las operaciones son completadas exitosamente sin fallas.

* + - 1. ***Consideraciones especiales***

Todo el software que no es objeto de prueba que es necesario y debe estar accesible.

* + 1. **Prueba de Instalación**
       1. ***Objetivo de la prueba***

Verificar que el software objeto de prueba se instala correctamente en cada configuración de hardware requerida bajo las siguientes condiciones:

* Instalar, una nueva máquina, nunca instalada previamente con Lybel
* Actualizar, máquina previamente instalada con Lybel, con la misma versión
  + - 1. ***Técnica***

Manualmente, para validar la condición de la máquina destino (nueva, nunca instalado, misma versión, versión anterior ya instalada).

Realizar la instalación.

Ejecutar un conjunto de pruebas funcionales ya implementadas para la Prueba de Funcionalidad.

* + - 1. ***Criterio de aceptación***

Cada prueba que se ejecute en Lybel debe pasar sin fallas de instalación.

* + - 1. ***Consideraciones especiales***
* Verificación de compatibilidad.
* Verificación de integridad.
* Configuración.
* Registro ante el dueño de la marca.
  + 1. **Prueba de Documentos**
       1. ***Objetivo de la prueba***

Verificar que el documento objeto de prueba sea:

* Correcto, esto es, que cumpla con el formato y organización para el documento establecido en el proyecto.
* Consistente, esto es, que el contenido del documento sea fiel a lo que hace referencia. Si el documento es Documentación de Usuario, que la explicación de un procedimiento sea exactamente como se realiza el procedimiento en el software, si se muestran pantallas que sean las correctas.
* Entendible, esto es, que al leer el documento se entienda correctamente lo que expresa y sin ambigüedades, además que sea fácil de leer.
  + - 1. ***Técnica***

Para verificar que el documento es correcto se debe comparar con el estándar definido si existe o con las pautas de documentación y ver que el documento cumple con ellas.

Para verificar que el documento es Consistente se debe ejecutar el programa siguiendo el documento en caso de los Materiales de Soporte al Usuario y comprobar que lo que se explica en estos documentos es exactamente lo que se ejecuta en el programa. En caso de Documentación Técnica se debe revisar el código al cual corresponde la documentación y comprobar que dicha describe el código.

Para verificar que el documento es entendible, debe comprobar que se entiende correctamente, que no tiene ambigüedades y que sea fácil de leer.

* + - 1. ***Criterio de aceptación***

El documento expresa exactamente lo que debe expresar, no hay diferencias entre lo que está escrito y el objeto de la descripción (operación de software, código de programa, decisiones técnicas) y se entiende fácilmente.

* + - 1. ***Consideraciones especiales***

N/A

* 1. **Herramientas**
* Junit– Testing unitario
* Github – Repositorio del proyecto
* NetBeans – IDE de desarrollo
* PgAdmin IV – Administrador de bases de datos PostgreSQL 12.3-1

1. **Recursos**

Aquí presentamos los recursos recomendados para el proyecto Ventas Lybel, sus principales responsabilidades y su conocimiento o habilidades.

* 1. **Roles**

En la tabla se muestra el personal en encargado de cada área para el proyecto Ventas Lybel en el área Verificación del Software.

|  |  |  |
| --- | --- | --- |
| **Rol** | **Cantidad mínima de recursos recomendada** | **Responsabilidades** |
| Responsable de verificación  (Fernando Figuereo) | **1** | Identifica, prioriza e implementa los casos de prueba.   * hacer Plan de Verificación. * Evalúa el esfuerzo para la  verificación. * hacer  Modelo de Prueba * Proporciona la dirección técnica. * obtener recursos Necesarios. * agregar informes de la verificación. |
| Asistente de verificación  (Wilson Gomez) | **1** | * Ejecuta las pruebas * Registra resultados de las pruebas. * Recuperar el software de errores. * Documenta los de cambio. * informes de prueba unitaria |
| Administrador de Base de Datos  (Fernando Figuereo , Wilson Gomez) | **2** | * Realiza mantenimiento del entorno de los datos (base de datos) de prueba y los recursos. * Administra la base de datos de prueba. |

* 1. **Sistema**

|  |  |
| --- | --- |
| **Recurso** | **Nombre/Tipo** |
| Servidor de base de datos | PostgreSQL 12.3-1 |
| Red o subred | -------- |
| Nombre del servidor | -------- |
| Nombre de la base de datos | VentasAdministracion |
| PC Cliente para pruebas | -------- |
| Requerimientos especiales | -------- |
| Repositorio de pruebas | https://github.com/FernandoRFR02/AplicacionJAVAFX |

1. **Hitos del proyecto de Verificación**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad que determina el hito** | **Esfuerzo**  **(Horas totales por semana)** | **Fecha de comienzo** | **Fecha de finalización** |
| Planificar la verificación | 40 | 11/05/2020 | 17/05/2020 |
| Elaborar casos de prueba | 20 | 20/05/2020 | 26/05/2020 |
| Ajuste y Control de Verificación | 16 | 27/05/2020 | 01/06/2020 |
| Ejecutar la verificación | 32 | 03/06/2020 | 14/06/2020 |
| Evaluar la verificación | 8 | 03/06/2020 | 14/06/2020 |

1. **Entregables**
   1. **Modelo de Casos de Prueba**

|  |  |
| --- | --- |
| Documento | **Modelo de Casos de Prueba** |
| Creado por | Verificación, equipo Ventas Lybel. |
| Para quién | Esta es la guía para realizar las pruebas del sistema y lo usarán los asistente de verificación. |
| Fecha de liberación | Será liberado el 27/05/2020. |

* 1. **Informes de Verificación**

|  |  |
| --- | --- |
|  |  |
| Creado por | Los desarrolladores que ejecutan las pruebas, equipo Ventas Lybel. |
| Para quién | Es el resultado para el equipo de  verificación, que detalla los errores encontrados para que puedan ser corregidos. |
| Fecha de liberación | luego de cada verificación unitaria. |

|  |  |
| --- | --- |
| Documento | se realizará un documento por cada prueba realizada. |
| Creado por | los encargado de ejecutar las pruebas el equipo de ventas Lybel. |
| Para quién | Los encargado de la documentación del proyecto para corregir los fallos encontrado. |
| Fecha de liberación | Será liberado luego de cada verificación de sistema. |

* 1. **Evaluación de la verificación**

|  |  |
| --- | --- |
| Documento | Se realiza un documento **Evaluación de la verificación** por cada prueba que se realice al sistema. Este contiene las fallas encontradas en el sistema. |
| Creado por | El Responsable de verificación, equipo Ventas Lybel. |
| Para quién | Es el resultado para el equipo de  verificación, que detalla los errores encontrados para que puedan ser corregidos. |
| Fecha de liberación | Será liberado luego de cada verificación, unitaria del sistema. |

* 1. **Informe final de verificación**

|  |  |
| --- | --- |
| Documento | El documento **Informe final de verificación** es el resumen de la verificación final del sistema antes de que sea liberado al entorno del usuario. |
| Creado por | El responsable de verificación, equipo Ventas Lybel. |
| Para quién | Desarrolladores |
| Fecha de liberación | Será liberado luego de la verificación final del sistema. |