**Plan de Pruebas**

**Portafolio de Título**

**Caso N°1**

**“CarWash”**

***[CarWash]***

***Fecha:[15/07/2022]***

**Tabla de contenido**

Histórico de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Descripción/cambio | autor |
| 1.0 | 15-07-2022 | Inicio de documento | Rodrigo Rodriguez |
|  |  |  |  |
|  |  |  |  |

Información del Proyecto

|  |  |
| --- | --- |
| Organización | Duoc UC. Escuela de Informática y Telecomunicaciones |
| Sección | 006-V |
| Proyecto (Nombre) | CarWash |
| Fecha de Inicio | 15-03-2022 |
| Fecha de Término | 14-07-2022 |
| Caso N° | 1 |
| Patrocinador principal | Sebastián Vidal |
| Docente | Sebastián Vidal |

Integrantes

|  |  |  |
| --- | --- | --- |
| Rut | Nombre | Correo |
| **18.975.147-8** | **Rodrigo Rodriguez** | **Rod.rodriguezf@duocuc.cl** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| Propósito del plan de pruebas  *Propósito, objetivo, visión que se espera de este plan de pruebas.* |
| El plan de pruebas tiene como objetivo orientar el esfuerzo de pruebas, identificando y detallando las pruebas más importantes, para que el equipo de QA pueda enfocarse en su ejecución y pueda responder de forma adecuada a los cambios que tiene el proyecto. |

|  |
| --- |
| Alcance de las pruebas  *Definición de requisitos de S.W., módulos de Software a probar, Requisitos ambiente de pruebas y Documentación Referenciada, etc.* |
| Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si tenemos una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin que nos demos cuenta. Los módulos para probar son:   * Registro * Login * Agenda cliente * Agenda administrador * Reportes * contratos |

|  |  |  |
| --- | --- | --- |
| Definición de roles y responsabilidades  *Roles y responsabilidades de todos los participantes en el* ***proceso de pruebas*** *de SW*. | | |
| Rol | Responsabilidades | Relevancia |
| Analista QA | Encargado del diseño de las pruebas |  |
| Testers | Encargado de ejecutar las pruebas |  |
| Desarrolladores | Pruebas unitarias |  |
|  |  |  |

|  |
| --- |
| Tipos de pruebas a realizar  *Definir el tipo de pruebas que se debe desarrollar para este proyecto, actividades y responsables.* |
| Las pruebas que se realizaron contemplan todo el sistema, ya que se ejecutaron pruebas para verificar el diseño del software, el funcionamiento a través de pruebas de componentes para verificar si los elementos que realizaban alguna acción funcionan correctamente, también se probó el funcionamiento del programa a través de lo que debería hacer, como el proceso de contrato de un servicio, el ingreso a la página a través de la autenticación, comprobar el funcionamiento de la base de datos, qué tipo de lenguaje y framework se utilizaron para poder desarrollar el sistema, y las pruebas que están dentro de las que se pueden listar son las siguientes:   * Pruebas unitarias: Las pruebas unitarias son las que aseguran que cada célula del código desarrollado en un componente brinde los resultados adecuados. En estas pruebas los desarrolladores observan la interfaz y la especificación de un componente, proporcionando la documentación del desarrollo del código se prueba exhaustivamente, claro que de forma independiente antes de pasar a otra unidad. * Pruebas de componentes: Las pruebas de componentes se ejecutan de forma independiente para comprobar que el resultado sea el requerido. Su objetivo es verificar las funcionalidades y/o usabilidades de los componentes, aunque no solo se limite a eso. * Pruebas de humo: Las pruebas de humo se realizan para verificar si las funcionalidades más significativas de la aplicación funcionan o no. De forma que lo más básico del software se ejecute de forma correcta con pruebas sencillas y rápidas. * Pruebas de integración: La prueba de integración es uno de los tipos de prueba funcional más común y se realiza de forma automatizada. Se realizan para probar componentes individuales con el objetivo de verificar cómo los módulos, que trabajan de forma individual, funcionan cuando estén integrados. |

|  |
| --- |
| Estrategia y técnicas de pruebas a aplicar  *Definir las estrategias y técnicas de pruebas que se debe desarrollar para este proyecto, actividades y responsables.* |
| Dependiendo de qué tipo de prueba sea y que es lo que se quiere abarcar, se implementarán estas que están diferenciadas en los dos grupos:  **PRUEBAS FUNCIONALES**   * Pruebas unitarias * Pruebas de componentes * Pruebas de humo * Pruebas de integración * Pruebas de regresión * Pruebas de cordura * Pruebas de aceptación   **PRUEBAS NO FUNCIONALES** |

|  |
| --- |
| Definición del proceso de testing  *Lisar y describir todas las actividades a desarrollar en el proceso general de testing, responsables, artefactos, etc.* |
| El encargado de realizar las actividades de testing es el testers rodrigo rodriguez y estas actividades son:   * Registrar usuarios * Login * Agregar servicio * Modificar servicio * Eliminar servicio * Contratar servicio * Cancelar contrato * Reset password * Exportar pdf * Exportar csv * Placeholder de los input |

|  |
| --- |
| Definición de ciclos de prueba a ejecutar  *Listar y describir cantidad de ciclos de prueba a realizar en este proyecto, las tareas y actividades para cada ciclo de prueba, responsables, artefactos, etc.* |
| las diferentes fases del ciclo de vida de las pruebas de software en detalle.  **Análisis de requisitos**  El testing de software debe ver, estudiar y analizar las especificaciones y requisitos disponibles. Ciertos requisitos producen resultados alimentándose con datos de entrada. En ese sentido, los probadores estudian los requisitos funcionales y no funcionales. Después de eso, deben elegir requisitos comprobables.  Las actividades son:   * lluvia de ideas para el análisis de requisitos e identificación y priorización de requisitos de prueba. * También incluyen la selección de requisitos para pruebas automáticas y manuales.   **Planificación de pruebas**  El segundo paso es la planificación de la prueba y el equipo de control de calidad crea este plan después de analizar todos los requisitos de prueba necesarios. Describen el alcance y los objetivos después de comprender el dominio del producto. Luego, el equipo analiza los riesgos involucrados y define los horarios y los entornos de prueba para crear una estrategia.  Después de eso, la gerencia finaliza las herramientas y asigna roles y responsabilidades a las personas. También se define un cronograma aproximado mediante el cual se deben completar las pruebas de cada módulo. Las actividades son:   * Investigar y recopilar posibles acciones sobre el producto. * Crea casos de prueba. * Priorice los casos de prueba. * Prepare scripts automatizados para casos de prueba.   **Configuración del entorno de prueba**  Las actividades de prueba necesitan ciertos factores ambientales, como servidores, marcos, hardware y software, para ejecutar casos de prueba desarrollados.  La configuración de software y hardware, junto con la configuración de datos de prueba, son los componentes principales de esta fase. Y es obligatorio realizar pruebas de humo y equipar a los evaluadores con herramientas de notificación de errores. Las actividades son:   * Comprender los requisitos mínimos. * Enumere el software y el hardware necesarios para diferentes niveles de rendimiento. * Priorice los entornos de prueba. * Configurar entornos de prueba. * Prueba de humo en los entornos construidos.   **Ejecución de la prueba**  Una aplicación está lista para probar una vez que el equipo haya terminado con todas las fases anteriores. Según el plan de prueba, los evaluadores ejecutan casos de prueba. También identifican, detectan y registran los defectos, informando así los errores. El equipo también es responsable de comparar los resultados esperados con el resultado real. Si se encuentra algún error, es necesario documentarlo para transmitirlo al equipo de desarrollo para que lo solucione. Las actividades son:   * Ejecute casos de prueba. * Identificar la desviación del comportamiento esperado del producto. * Registre casos fallidos con detalles. * Pruebe de nuevo después de corregir los errores.   **Prueba de cierre**  El final de la ejecución de la prueba y la entrega del producto final marca el inicio de la fase de cierre de la prueba. El equipo de control de calidad verifica los resultados de la prueba y los analiza con otros miembros del equipo. Algunos otros factores que consideran son la calidad del producto, la cobertura de la prueba y el costo del proyecto. Si hay una desviación de los valores estimados, se pueden realizar más análisis para identificar lo que no salió como se esperaba. Las actividades son:   * Verificar que se completen todas las pruebas. * Evaluar factores como la calidad, la cobertura de la prueba, el cronograma y el costo. * Documentar la conclusión. * Discutir el aprendizaje y averigüe si se puede mejorar el proceso de prueba. * Preparar el informe de cierre de la prueba. |

|  |
| --- |
| Calendarización de las actividades de pruebas  *Listado de actividades, tareas, duración, fechas, responsables, etc.* |
| [Carpetas\semestre\_5\PROYECTO\Carta Gantt (1).xlsx](Carpetas/semestre_5/PROYECTO/Carta%20Gantt%20(1).xlsx) |
| Adjuntar carta Gantt |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Resumen de riesgos  *Listado de riesgos relacionado al proceso de pruebas de S.W. Indicar riesgo, magnitud o impacto de este riesgo por etapa en el proceso. Magnitud: Alto , Significativo , Moderado, Inferior y Baja.* | | | | | | |
|  | **Fase del proceso de pruebas** | | | | | **Riesgo** |
| **Planificación** | **Análisis y diseño** | **Implementación y ejecución** | **Evaluación** | **Cierre** |
| **Magnitud** |  |  |  |  |  |
| **No crear caso de prueba de algún modulo** |  |  |  |  | moderado |
|  | **Que quede fuera algún requisito funcional** |  |  |  | Significativo |
|  |  | No registrar casos fallidos |  |  | inferior |
|  |  |  |  | desviación de los valores estimados | Alto |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| Clasificación de los defectos  *Definir la clasificación de los defectos según su nivel de severidad* | |
| Nivel de Severidad | Descripción |
| Medio | La verificación de validación en el ingreso de días correspondientes al mes no está funcional. |
| Medio | La verificación de validación en el ingreso de los meses correspondientes al año no está funcional. |
| mejora | La validación de la separación de miles para los precios no se ve reflejada |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Definición de artefactos  *Listar y describir los artefactos que serán administrados y entregados durante este proceso de prueba.* | |
| Artefacto | Descripción |
| Planilla plan de pruebas | Planilla donde se muestra de forma detalla las pruebas realizadas al proyecto. |
| Planilla registro de defectos | Planilla donde se muestra de forma detallada los defectos encontrados en el software, severidad y estado en el que se encuentran. |
|  |  |
|  |  |

|  |
| --- |
| Condiciones de aceptación para cierre del proceso de pruebas  *Condiciones que se deben cumplir para dar termino al proceso de pruebas y margen de tolerancia de aceptación de defectos.* |
| Las funcionalidades de los módulos Registro, login, agenda cliente, agenda administrador, reportes, contratos deben estar 100% funcional. Y que los defectos el nivel de severidad sea inferior a grave. |