| **PROYECTO** | **Sistema de gestión de alquileres de casas** | **CÓDIGO DE PROYECTO** | **2024-100** | **FECHA DE ELABORACIÓN** | **17/10/2024** |
| --- | --- | --- | --- | --- | --- |

**NORMAS Y PROCEDIMIENTOS A APLICAR**

| * Informes de seguimiento durante la ejecución: Garantizar que el proyecto avance de acuerdo con el cronograma y los requisitos establecidos, y que el patrocinador esté al tanto del progreso.   Se realizarán informes de seguimiento semanales para revisar los avances en la implementación de funcionalidades y  solucionar posibles bloqueos.   * Control de versiones mediante git: Para asegurar una correcta gestión del código fuente y facilitar la fase de ejecución del proyecto. Las políticas de ramas, revisión de código y convenciones de mensajes de commits.   *develop*: Esta es la rama de desarrollo continuo. Toda la funcionalidad nueva que se va implementando se desarrolla en ramas derivadas de develop. De la rama develop salen ramas de características, también llamadas "paquetes de trabajo". Estas ramas están dedicadas a una tarea o funcionalidad específica, lo que ayuda a aislar el trabajo de un desarrollador sin afectar el código principal. Estas ramas tendrán nombres como feature/nombre\_módulo.  Una vez que una funcionalidad está terminada y probada, se hace un merge de vuelta a develop para asegurar que las nuevas funcionalidades se integren y estén disponibles para el resto del equipo.  A medida que las tareas o características se completan, se fusionan de nuevo en develop para integrarse con el resto del proyecto.  *docs*: Es una rama especializada para los documentos del proyecto. Este flujo te permite manejar la documentación del producto sin interferir con el código fuente ni las funcionalidades de la aplicación.  Cuando el proyecto está listo para una nueva versión o entrega al cliente, *release* se convierte en la rama final donde se agrupan todas las funcionalidades terminadas. Aquí se realiza una última fase de pruebas y ajustes finales. En esta etapa, los errores o problemas menores se solucionan antes de realizar el merge final.  Para las revisiones de código se realizará a través de Pull Requests, y al menos un miembro del equipo deberá revisar cada PR antes de realizar un merge a cualquier rama.  Además, has definido una política para los commits donde se utiliza el formato feat/fix/docs:"*mensaje*", con el mensaje redactado en tiempo imperativo.   * Reuniones semanales con el patrocinador: Se realizarán reuniones semanales con el patrocinador para recibir feedback de las nuevas implementaciones en el proyecto. * Pruebas unitarias y de integración que cubran todas las funcionalidades del código: Se realizarán estas pruebas con el fin de garantizar que el código funciona correctamente tanto en piezas individuales como en la interacción entre los diferentes componentes. |
| --- |

| **OBJETIVOS DE CALIDAD** | | |
| --- | --- | --- |
| **ENTREGABLE** | **MÉTRICA** | **VALOR OBJETIVO** |
| Producto | Variación del cronograma | ≤ 5% de variación |
| Producto | Desviación del cronograma | ≤ 5% de desviación |
| Producto | Porcentaje de casos de prueba aprobados | ≥ 95% aceptado |
| Producto | Porcentaje de casos de prueba fallidos | ≤ 5% fallidos |
| Producto | Porcentaje de defectos críticos | porcentaje de 0% defectos críticos |
| Producto | Porcentaje de defectos diferidos | ≤ 5% diferidos |

**ACTIVIDADES DE CONTROL DE CALIDAD DE LOS ENTREGABLES**

| **EDT #** | **NOMBRE PAQUETE DE TRABAJO** | **ACTIVIDAD** | **RESPONSABLE** |
| --- | --- | --- | --- |
| 3.1.1 | Registro de usuarios | Analizar la desviación y variación del cronograma | Equipo de desarrollo |
| 3.1.2 | Gestión de propiedades | Analizar la desviación y variación del cronograma | Equipo de desarrollo |
| 3.1.3 | Búsqueda y filtrado de pisos | Analizar la desviación y variación del cronograma | Equipo de desarrollo |
| 3.1.4 | Reserva de pisos | Analizar la desviación y variación del cronograma | Equipo de desarrollo |
| 3.1.5 | Gestión de pisos reservados | Analizar la desviación y variación del cronograma | Equipo de desarrollo |
| 3.1.6 | Implementación de notificaciones | Analizar la desviación y variación del cronograma | Equipo de desarrollo |
| 3.2.1 | Pruebas de funcionalidad | Analizar el porcentaje de casos de pruebas fallidos y aprobadas y el porcentaje de defecto críticos y diferidos. | Equipo de desarrollo |
| 3.2.2 | Pruebas de seguridad y aceptación | Analizar el porcentaje de casos de pruebas fallidos y aprobadas y el porcentaje de defecto críticos y diferidos. | Equipo de desarrollo |
| 4.1 | Incidencias y seguimiento | Analizar la desviación y variación del cronograma | Equipo de desarrollo |

| **LISTAS DE CONTROL** | | |
| --- | --- | --- |
| **#** | **DURACIÓN** | **DESCRIPCIÓN** |
| 1 | 1 día | Ver que pasan todas las pruebas de seguridad. |
| 2 | 1 día | Ver que pasen todas las pruebas unitarias y de integración. |

**PLAN DE MEJORA DE LOS PROCESOS DE GESTIÓN**

| **ACTIVIDAD #** | **NOMBRE DE LA ACTIVIDAD** | **METODOLOGÍA O PROCEDIMIENTO** | **RESPONSABLE** |
| --- | --- | --- | --- |
| 1 | Planificar | Se definirán pruebas piloto con:  La actividad a mejorar.  Los objetivos a alcanzar.  Los indicadores de control (métricas).  Los métodos y herramientas para  capturar los indicadores de control. | Equipo de dirección |
| 2 | Hacer | Llevaremos a cabo el plan anterior sobre las actividades señaladas, realizando los cambios en los procesos o procedimientos de las actividades y recopilando datos. | Equipo de desarrollo |
| 3 | Verificar | Comprobamos si las mejoras implantadas han alcanzado los objetivos. | Equipo de desarrollo |
| 4 | Actuar | Si la verificación ha tenido éxito, transformamos la mejora en una norma o procedimiento y lo  incorporamos a las lecciones aprendidas para que pasen luego a los APO. En caso contrario,  examinaremos los experimentos e iniciaremos un nuevo ciclo. | Equipo de dirección |

| **HITOS A INCLUIR** | | |
| --- | --- | --- |
| **#** | **DURACIÓN** | **DESCRIPCIÓN** |
| 1 | 1 día | Mejora estandarizada. |

| **REUNIONES A REALIZAR** | | |
| --- | --- | --- |
| **#** | **DURACIÓN** | **DESCRIPCIÓN** |
| 1 | 1 día | Se llevarán a cabo auditorías semanales con todo el equipo de desarrollo para evaluar el progreso del proyecto en cada iteración, realizando una retrospectiva para garantizar que todas las tareas se estén ejecutando de acuerdo con los estándares establecidos. |
| 2 | 1 día | Además, se llevará a cabo una reunión dedicada a recopilar y documentar las lecciones aprendidas al finalizar el proyecto, lo que ayudará a mejorar procesos y enfoques en futuros desarrollos. |

| **INFORMES A ELABORAR** | | |
| --- | --- | --- |
| **#** | **DURACIÓN** | **DESCRIPCIÓN** |
| 1 | 1 día | Por cada día de trabajo del equipo de desarrollo, se elaborará un informe de desempeño que incluirá detalles sobre las tareas completadas, el tiempo invertido en cada actividad y cualquier comentario relevante. Este informe permitirá monitorear el progreso del proyecto, identificar áreas de mejora y garantizar la transparencia en el desempeño del equipo. |
| 2 | 1 día | Por cada iteración realizará un informe de seguimiento del proyecto y de costes, esto nos permitirá tomar decisiones según datos actualizados. |

| **AUDITORÍAS EXTERNAS** | | |
| --- | --- | --- |
| **#** | **DURACIÓN** | **DESCRIPCIÓN** |
| 1 | 1 día | Se programará una reunión semanal con el patrocinador para garantizar que el equipo de desarrollo esté alineado con las expectativas del cliente y que el avance del proyecto cumpla con sus requisitos y objetivos. |

**HERRAMIENTAS PARA LA GESTIÓN DE LA CALIDAD**

| - Git: Permite llevar un registro de los cambios en el código, facilitando la colaboración y la revisión de código.  MSProject.  - MSProject: Esta herramienta nos permitirá planificar, programar y hacer seguimiento de las actividades e hitos del proyecto de manera eficiente. |
| --- |