**Introducción**

El propósito de este documento es servir como guía de referencia que permita comprender la estructura y el funcionamiento de cada elemento de la aplicación web desarrollada durante el curso de "Diseño de Aplicaciones Web". Aquí se proporcionan términos clave que se utilizan en el proyecto para diferenciar entre los distintos componentes de manera específica. Además, se brinda una descripción detallada de todas las interacciones disponibles en el sistema, con el objetivo de clarificar la lógica de funcionamiento en la medida de lo posible.

En las próximas páginas de este documento, se presentan de forma clara y concisa las características del sistema, con el objetivo de hacerlo comprensible para cualquier lector. Las siguientes secciones describen de manera sencilla cómo opera el sistema, de manera que su funcionamiento pueda ser entendido por cualquier persona que lo consulte.

**Alcance y propósito**

**Objetivos**

El proyecto "Paseos El Puerto" es un sistema de gestión y reserva de paseos en embarcaciones. Su objetivo principal es funcionar de intermediario entre los propietarios de embarcaciones y los clientes, lo que ayuda a maximizar el uso de las embarcaciones, para sacarles el mayor provecho posible. El sistema permite la organización, programación y disfrute de paseos. Además, el sistema ofrece funcionalidades para gestionar el estado de las embarcaciones, como el mantenimiento y las reparaciones, lo que garantiza que las embarcaciones estén en condiciones adecuadas para los paseos y minimiza los inconvenientes para los clientes.

**Funciones principales**

* Gestión de embarcaciones
* Gestión de paseos
* Seguimiento de mantenimiento y reparaciones
* Gestión de propietarios
* Asociación de embarcaciones
* Gestión de pagos y cobros
* Registro de clientes

**Aspectos de funcionamiento**

* Gestión de embarcaciones: Permite registrar, mostrar, eliminar, buscar y modificar información sobre las embarcaciones disponibles o que no están disponibles en el puerto, dicha información incluye un identificador, el modelo, año, la fecha de inicio del contrato y la fecha de fin, el nombre y el costo por hora de la embarcación.
* Gestión de paseos: Permite programar y administrar los paseos en las embarcaciones, incluyendo la duración, que es la diferencia entre las fechas de inicio y fin y el cálculo del monto total, contemplando si el cliente necesita personal o no para dicho paseo.
* Seguimiento de mantenimiento y reparaciones: Permite registrar y gestionar actividades de mantenimiento y reparaciones realizadas en las embarcaciones, asegurando su buen estado y cumplimiento de normas de seguridad.
* Gestión de propietarios: Permite registrar, mostrar, eliminar, buscar y modificar información sobre los propietarios de las embarcaciones, dicha información incluye sus datos personales, información de contacto y detalles relacionados con la propiedad de las embarcaciones.
* Asociación de embarcaciones: Permite asociar las embarcaciones registradas en el sistema con sus respectivos propietarios. Esto facilita la identificación y seguimiento de las embarcaciones y su propiedad.
* Gestión de pagos y cobros: Proporciona funcionalidades para administrar los pagos y cobros relacionados con la propiedad de las embarcaciones. Esto puede incluir el seguimiento de pagos de alquiler o comisiones generadas por los paseos realizados.
* Gestión de clientes: Permite al usuario registrar, mostrar, eliminar, buscar y modificar información de los clientes en el sistema, dicha información incluye de tipo personal, datos de contacto y cualquier otra información relevante. Esto facilita la identificación y seguimiento de los clientes.

**Aspectos por implementar**

* Registrar y autenticar usuarios para personalizar la experiencia del usuario.
* Hacer un seguimiento de reservas, que los clientes puedan acceder a un registro de sus reservas anteriores y futuras, lo que les permita realizar un seguimiento de sus actividades y realizar cambios o cancelaciones si es necesario.
* Administración de embarcaciones al alcance de los propietarios.
* Generación de informes y estadísticas sobre diferentes aspectos del negocio, como la cantidad de reservas, ingresos generados, popularidad de los paseos, entre otros, para ayudar en la toma de decisiones y el seguimiento del rendimiento.
* Formas de ayuda para los clientes, ya sea por medio de correos o por medio de un chat de asistencia rápida dentro del sistema.
* Gestión de cancelaciones y reembolsos, para permitir a los clientes cancelar sus reservas y solicitar reembolsos en caso de que sea necesario. Esto puede estar sujeto a políticas específicas definidas por el negocio.
* Comunicación y notificaciones de envío de confirmaciones de reserva, recordatorios de paseos, actualizaciones de eventos o promociones especiales.
* Valoraciones y comentarios, para permitir a los clientes proporcionar valoraciones y comentarios sobre su experiencia en los paseos. Esto puede ayudar a mejorar la calidad de los servicios y proporcionar información útil a otros posibles clientes.

**Restricciones**

* Utiliza Java como lenguaje de programación.
* Utiliza PostgreSQL como motor de base de datos.
* El sistema está diseñado sin contemplar la concurrencia, lo que significa que solamente puede ocuparse cada apartado por un único usuario, pueden ocurrir problemas si se hace más de un registro de una misma entidad al mismo tiempo.

**Planificación**

**Agenda**

Durante la última semana de abril (24 al 30 de abril de 2022):

* El 26 de abril se planteó el problema.
* El 29 de mayo se diseñó el diagrama E-R de la base de datos preliminar, definiendo entidades y relaciones.

En la segunda semana de mayo (8 al 14 de mayo):

* El 14 de mayo se creó el repositorio y se desarrollaron los JavaBeans preliminares.

En la tercera semana de mayo (15 al 21 de mayo):

* El 18 de mayo se creó una clase para establecer comunicación con la base de datos.
* El 19 de mayo se empezó a trabajar en la creación de la entidad “Clientes”.
* El 21 de mayo se creó la primer clase DAO, la de la entidad "Clientes”.

En la cuarta semana de mayo (22 al 28 de mayo):

* El 22 de mayo se creó el primer Servlet, el de la entidad "Clientes”.
* El 23 de mayo se empezó a trabajar en las vistas, principalmente en el index.jsp.
* El 24 de mayo se terminaron las vistas de la entidad “Clientes”
* El 25 de mayo se finalizó el funcionamiento de la Entidad “Clientes”, lo cuál sirvió como plantilla para las demás entidades.
* El 26 de mayo se trabajó en los estilos de las vistas.
* El 27 de mayo se finalizó el funcionamiento de la Entidad “Propietarios”.
* El 28 de mayo se empezó a trabajar en las “Embarcaciones”.

En lo que resta de mayo (29 al 31 de mayo):

* El 29 de mayo se reajusto el diseño de la base de datos, quitando y agregando entidades como la de “Contratos”
* El 30 de mayo se empezó a trabajar en la entidad “Contratos” paralelamente con “Embarcaciones”.
* El 31 de mayo se finalizó el funcionamiento de las Entidades “Contratos” y “Embarcaciones”

En la primera semana de junio (1 al 7 de junio):

* El 1 de junio se empezó a trabajar con la entidad “Paseos”.
* El 3 de junio se empezó a trabajar en la entidad “Paseos\_Personal” paralelamente con “Paseos”.
* El 4 de junio se realizaron cambios finales al diseño de la base de datos.
* El 5 de junio se empezó a trabajar con la entidad “Personal” paralelamente a “Paseos”.
* El 7 de junio se creó una vista para el monto total del paseo.
* En la segunda semana de junio (8 al 14 de junio):
* El 9 de junio se finalizó el funcionamiento de las entidades “Paseos”, “Paseos\_Personal” y “Personal”, incluyendo el correcto funcionamiento de la vista “Monto\_total.jsp”.
* El 10 de junio se empezó a trabajar en la entidad “Mantenimiento” y se terminó el mismo día en la noche
* El 11 de junio se empezó con la entidad “Reparación” y se terminó el mismo día en la tarde, en la noche a las 9:30pm aproximadamente di por concluido el proyecto, para luego empezar la documentación.

**Análisis y diseño del sistema**

**Metodología utilizada**

Durante la ejecución del proyecto Paseos "El Puerto", se adoptó la metodología ágil conocida como Scrum, la cual se seleccionó debido a sus ventajas en términos de agilidad y calidad en el desarrollo del software. Scrum se basa en la idea de dividir el proyecto en ciclos de trabajo llamados sprints, donde cada sprint tiene una duración máxima de un mes. Esta elección se hizo en consideración de la necesidad de adaptación a los posibles cambios y requisitos cambiantes del proyecto, y para fomentar la entrega de valor de forma incremental.

En el contexto específico de este proyecto, se identificaron tres roles principales en la metodología Scrum:

Product owner: Este rol recae en un miembro del equipo encargado de mantener una comunicación fluida con el cliente. Su responsabilidad radica en mostrar los avances del proyecto al cliente y recibir retroalimentación valiosa. En este caso, el product owner puede formar parte del equipo de desarrollo mismo.

Scrum máster: Este rol es asignado a un miembro del equipo que tiene la tarea de garantizar que todos los miembros comprendan y apliquen correctamente las técnicas de Scrum. El Scrum máster se enfoca en eliminar los obstáculos o dificultades que puedan surgir durante el sprint, asegurando un flujo de trabajo eficiente y sin interrupciones.

Equipo de desarrollo: Este grupo está compuesto por los miembros encargados de desarrollar el software en sí. Su responsabilidad principal es llevar a cabo las tareas prioritarias definidas por el product owner y trabajar en colaboración para lograr los objetivos establecidos para cada sprint.

La metodología Scrum se caracteriza por fomentar la transparencia, la colaboración y la adaptabilidad. Los sprints, al establecer plazos específicos y objetivos claros, permiten al equipo enfocarse en tareas concretas y obtener una retroalimentación regular del cliente. Esta flexibilidad inherente a Scrum brinda la capacidad de realizar ajustes en el proyecto según sea necesario, con el fin de cumplir con las expectativas del cliente y alcanzar los objetivos establecidos de manera efectiva y eficiente.

**Diagramas UML**

***Diagrama general del sistema***

**Texto

Descripción generada automáticamente*Diagramas de clases***

*CLASES DAO*

**Imagen que contiene camino, computadora, alambre, tabla

Descripción generada automáticamente**

*CLASES JAVA BEANS*

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

*CLASE DE CONEXIÓN A LA BASE DE DATOS*

***Diagramas de casos de uso***