



DESAFIO DE PROGRAMACIÓN

TICSUR 2024

REGLAMENTO Y CASO PARA DESARROLLAR

Contenido

1.	DESCRIPCIÓN DEL CASO.....	3
2.	REQUISITOS DE SOFTWARE	4
2.1.	Requisitos Funcionales.....	4
2.2.	Requisitos No Funcionales.....	5
3.	ETAPAS DEL DESAFÍO.....	6
	Etapa 1: Planificación y Especificación Inicial.....	6
	Etapa 2: Diseño Detallado	6
	Etapa 3: Diseño de la Base de Datos y Desarrollo	7
	Etapa 4: Implementación, Prototipo y Evaluación	7
	Etapa 5: Presentación Final y Documentación	7
4.	REGLAS DEL DESAFÍO DE PROGRAMACIÓN	8
4.1.	Recursos Permitidos	8
4.2.	Independencia del Equipo.....	8
4.3.	Ejecución en Paralelo.....	8
4.4.	Horario y Entregas.....	8
4.5.	Uso de Datos y Seguridad	8
4.6.	Conducta Ética.....	9
4.7.	Criterios de Evaluación	9
4.8.	Premios y Reconocimientos	9
4.9.	Documentación.....	9

1. DESCRIPCIÓN DEL CASO

En la Universidad e Instituto profesional Santo Tomás, así como en la ciudad de Temuco y otros lugares, cada vez más personas eligen la bicicleta, scooters, patinetas y otros como medio de transporte para desplazarse al trabajo, estudiar o hacer ejercicio. Debido a este creciente número de usuarios y a la poca disponibilidad de lugares para estacionarse tanto en la institución como en la ciudad, se plantea la necesidad de contar con una gestión más eficiente de los espacios de estacionamiento para estos medios de movilidad alternativos. Aunque este aumento es beneficioso para la salud y el medio ambiente, también genera problemas significativos para los usuarios a la hora de buscar estacionamiento o un lugar donde dejar de forma segura su medio de transporte alternativo. Problemáticas como el desorden en los estacionamientos y el incremento de robos se agravan cuando las medidas actuales, como el uso de dobles candados, no son soluciones efectivas y consumen tiempo valioso tanto para los alumnos como para los guardias.

Para abordar estos desafíos, surge la idea de desarrollar un sistema de gestión de estacionamientos para medios alternativos de transporte urbano. Los usuarios principales de este sistema serán los alumnos, y existirán roles específicos para administrar los espacios de estacionamiento, facilitando así la organización y el uso eficiente de los mismos. Se anima a los alumnos a proponer soluciones innovadoras, de bajo costo y con un alto impacto en la resolución de estos problemas.

El sistema debe ser independiente y no requerirá integración con otros sistemas existentes, aunque deberá ser capaz de interactuar con el GPS del móvil para localizar espacios disponibles. En cuanto al presupuesto, se busca que el costo sea el menor posible, sin regulaciones legales específicas que limiten su implementación. Además, es fundamental que el sistema sea escalable para permitir futuras expansiones a otras sedes de la universidad en diferentes ciudades, y por el momento no se considerará el mantenimiento continuo del sistema.

2. REQUISITOS DE SOFTWARE

2.1. Requisitos Funcionales

A. Registro y Autenticación de Usuarios

- Los usuarios (alumnos) deben poder registrarse e iniciar sesión en el sistema.
- Debe haber roles diferenciados para los usuarios, incluyendo administradores de estacionamiento y usuarios regulares.

B. Visualización y Búsqueda de Espacios

- Los usuarios deben poder visualizar en tiempo real los espacios disponibles para estacionar sus medios de transporte.
- Debe existir una función de búsqueda de espacios disponibles ya sea basada en la ubicación actual del usuario, utilizando el GPS del móvil, o mediante una lista por edificio (sede Rodríguez, Sede Pedro de Valdivia, etc).

C. Reserva de Espacios

- Los usuarios deben poder reservar un espacio de estacionamiento en uno o más bloques horarios preconfigurados. Ejemplo (de 08:00-09:59 | de 10:00 – 11:59 | etc).
- Los bloques horarios de uso diario pueden ser configurados por los usuarios administradores.
- El sistema debe notificar al usuario cuando el tiempo de uso esté por expirar (según el bloque), ofreciendo la opción de extenderla.

D. Gestión de Espacios por Administradores

- Solo los administradores tendrán acceso a la gestión de espacios. Las opciones de añadir, modificar o eliminar espacios de estacionamiento, así como gestionar las reservas, no estarán disponibles para los usuarios regulares.
- Deben poder manejar conflictos o necesidades especiales relacionadas con las reservas, las cuales son notificadas por los usuarios.

E. Notificaciones

- Los usuarios deben recibir notificaciones sobre el estado de su reserva (confirmación, expiración, entrega).

F. Reporte de Incidentes y Uso de Espacios

- Los usuarios deben poder reportar problemas o incidentes relacionados con los estacionamientos (por ejemplo, robos o daños).
- Deben informar al sistema cuando desocupen un espacio de estacionamiento. *En este punto se anima a los estudiantes a proponer soluciones innovadoras como el uso de códigos QR, tecnología RFID, o sensores para detectar automáticamente cuando un espacio se desocupa.*

2.2. Requisitos No Funcionales

A. Usabilidad

- Interfaz de usuario intuitiva y fácil de usar en dispositivos móviles y/o en la web.

B. Rendimiento

- Respuestas rápidas del sistema, especialmente en la visualización y actualización de los espacios disponibles.

C. Seguridad

- Protección de datos personales y financieros de los usuarios.
- Autenticación segura y manejo de sesiones para prevenir accesos no autorizados.

D. Escalabilidad

- Capacidad para manejar un aumento en el número de usuarios y estacionamientos sin degradar el rendimiento.
- Flexibilidad para expandir a más ciudades o incorporar otros tipos de vehículos.

E. Costo

- Implementación y operación del sistema con el menor costo posible, optimizando recursos y tecnología.

3. ETAPAS DEL DESAFÍO

Para estructurar el desarrollo del caso de estudio en el desafío de programación, se dividió en varias etapas clave. Cada una de estas etapas representará un logro con una puntuación determinada y tendrá entregables específicos que permitirán evaluar el progreso de los equipos. A continuación, se describen las etapas y sus entregables:

Etapa 1: Planificación y Especificación Inicial

Objetivo: Establecer rápidamente el alcance y los requisitos del proyecto.

Duración estimada: 30 min.

Puntaje: 10 pts.

Entregables:

- **Definición de Roles y Metodología de Trabajo:** Breve descripción que aclare cada uno de los roles de los integrantes, el alcance del proyecto y cómo abordarán el desarrollo. Este documento debe ser conciso para permitir una rápida comprensión y puesta en marcha del desarrollo.
- **Selección Rápida de Herramientas y Tecnologías:** Decisión inmediata sobre las herramientas y tecnologías a utilizar, basándose en la familiaridad del equipo y la adecuación para las tareas requeridas.

Etapa 2: Diseño Detallado

Objetivo: Crear los diseños arquitectónicos y detallados del sistema.

Duración estimada: 1:30 Hrs.

Puntaje: 30 pts.

Entregables:

- **Diseño de Arquitectura del Sistema:** Propuesta arquitectónica del sistema desarrollada por el equipo, describiendo la estructura de módulos, hardware y tecnologías a utilizar.
- **Diagramas UML:** Incluyendo diagramas de casos de uso, diagramas de clases, diagramas de secuencia, diagramas de flujos o procesos y diagramas de estados.
- **Mockups de la Interfaz de Usuario:** Diseños preliminares de la interfaz de usuario que muestran la navegación y la interacción básica del sistema.

Etapa 3: Diseño de la Base de Datos y Desarrollo

Objetivo: Definir la estructura de datos que soportará el sistema y desarrollar la base de datos.

Duración estimada: 2 Hrs.

Puntaje: 40 pts.

Entregables:

- Modelo Entidad-Relación (MER): Diseño conceptual de la base de datos.
- Diagrama de Relación de Entidades (DER): Representación detallada de la base de datos que incluye tablas, atributos y relaciones.
- Script de la Base de Datos: Script SQL para la creación de la base de datos, incluyendo tablas, relaciones y posiblemente algunos datos iniciales.

Etapa 4: Implementación, Prototipo y Evaluación

Objetivo: Desarrollar un prototipo funcional y realizar pruebas iniciales.

Duración estimada: 2 Hrs.

Puntaje: 50 pts.

Entregables:

- Código Fuente: Implementación de la funcionalidad básica del sistema, enfocándose en un prototipo rápido que demuestre la viabilidad del proyecto.
- Informe de Testing: Documentación de los tests realizados, incluyendo descripción de pruebas, resultados obtenidos y posibles mejoras.
- Documentación Completa del Proyecto en GitHub: Incluyendo todos los diagramas UML actualizados, código fuente final, scripts de base de datos almacenados y documentados en un repositorio de GitHub.

Etapa 5: Presentación Final y Documentación

Objetivo: Presentar el sistema completo y entregar toda la documentación requerida.

Entregables:

- Presentación Final del Proyecto: Una demostración completa de la ejecución de la actividad y de los distintos entregables. Opcionalmente mostrar el MVP (Producto mínimo viable) funcionando si se logró desarrollar.

4. REGLAS DEL DESAFÍO DE PROGRAMACIÓN

Las siguientes reglas están diseñadas para fomentar un ambiente competitivo pero justo, donde los participantes puedan demostrar sus habilidades técnicas y creativas bajo un marco de integridad.

4.1. Recursos Permitidos

- Los participantes pueden buscar información en internet durante el desafío.
- Está permitido el uso de herramientas de inteligencia artificial y sitios web para la creación de diagramas y diseño de interfaces.
- Se pueden utilizar todos los IDEs, librerías y frameworks que consideren necesarios, siempre y cuando sean de acceso libre o estén previamente aprobados por los organizadores.

4.2. Independencia del Equipo

- Los equipos no deben recibir ayuda externa de personas que no estén oficialmente registradas en el equipo.
- La colaboración o asistencia entre diferentes equipos no está permitida.

4.3. Ejecución en Paralelo

- Las etapas del proyecto se pueden trabajar en paralelo. Los equipos pueden distribuir tareas como consideren apropiado para maximizar la eficiencia y efectividad.

4.4. Horario y Entregas

- Todos los equipos deben adherirse al horario establecido para la competencia, que concluirá con una presentación y entrega final de todos los entregables.
- El horario límite para entregar y presentar el proyecto será estrictamente respetado.
- Los proyectos entregados después del horario establecido no serán evaluados.

4.5. Uso de Datos y Seguridad

- Cualquier uso de datos reales dentro de las aplicaciones debe cumplir con las normativas de privacidad y protección de datos aplicables. Se recomienda utilizar datos ficticios o anónimos para el desarrollo y las demostraciones.

4.6. Conducta Ética

- Se espera que todos los participantes mantengan una conducta ética a lo largo del evento, respetando las reglas y a los demás participantes.
- Cualquier forma de plagio o copia de trabajo ajeno resultará en la descalificación inmediata del equipo.

4.7. Criterios de Evaluación

- Los proyectos serán evaluados en base a la originalidad, funcionalidad, complejidad técnica, cumplimiento de los requisitos, calidad del código y diseño, y la eficacia de la presentación final.
- Los jueces pueden realizar preguntas durante las presentaciones para entender mejor las decisiones de diseño y la implementación.

4.8. Premios y Reconocimientos

- Se otorgarán premios a los equipos que logren los mayores puntajes y se destaquen en aspectos específicos del desarrollo como mejor diseño, mejor uso de tecnología, y solución más innovadora.

4.9. Documentación

- Todos los equipos deben mantener una documentación adecuada de su proyecto en un repositorio de GitHub, que será revisado como parte de la evaluación final.