|  |  |
| --- | --- |
|  | UNIDADES TECNOLOGÍCAS DE SANTANDER |
| **Programa** | Tecnología en Desarrollo de Sistemas de Información |
| **Curso** | Webservice |
| **Docente** | Msl Ing. Camilo Ernesto Rodriguez Moreno |
| **Fecha de entrega:** | 30/04/2022 |
| **Nota** | Este trabaja tiene un valor del 60% y el trabajo práctico del 40% |

|  |
| --- |
| **TRABAJO I** |
| 1. **Título del proyecto.**   Aplicación web para la gestión automatizada de parqueaderos AppPark.   1. **Resumen.**   En el siguiente proyecto se busca agilizar el proceso administrativo de un parqueadero con respecto al control de los cobros de las tarifas de acuerdo al tiempo de duración de parqueo de cada tipo de vehículo. Es por esta razón que esta problemática se busca optimizar este proceso mediante un registro de ingreso, salida de cada vehículo y su cobro que está determinado de acuerdo a la categoría que pertenezca (automóvil, camioneta, motocicleta).  Cada tipo de vehículo tiene estandarizado un cobro según el tiempo de parqueo, además se tiene un registro de cada bahía y su estado actual, es decir si se encuentre ocupado o no, esto con el fin de agilizar la asignación de parqueo para cada tipo de vehículo.   1. **Objetivo General.**   Desarrollar un software que permita el control administrativo de parqueo utilizando SpringBoot y PostgreSQL.   1. **Justificación.**   Actualmente, la transformación digital es considerada como uno de los factores innovadores que combinan la tecnología con el trabajo repetitivo, para facilitar el desarrollo de actividades, ahorrando tiempo y reprocesos.  Para ello, se requieren las mejoras necesarias en el parqueadero en las que se pueda controlar con precisión el flujo de tráfico que se cuentan en el parqueadero. Un mejor control del proceso garantizará una mejor organización. Con la ayuda de este software se garantiza una mejor gestión y organización de la contabilidad, control y gestión, dando solución a los procesos manuales que se realizan actualmente.     1. **Diagrama de clases**        1. **Prototipo**      * + Construir el API REST con spring boot y hacer el deploy en heroku.   + Fuera de los cruds que puedan surgir debe tener mínimo dos servicios específicos del negocio. Por ejemplo, una app para biblioteca de una universidad para el préstamo de libros para estudiantes. Se tienen las siguientes clases con sus relaciones: estudiante, libro, ejemplar, autor, de la cuales vamos a tener algunos cruds con sus servicios, y adicional los servicios del negocio, como por ejemplo, prestar ejemplar o reservar ejemplar. El prototipo debe tener uno o dos servicios del negocio. |
|  |