|  |  |  |
| --- | --- | --- |
|  |  | |
| ***INFORMÁTICA*** | | |
| ***GRADO SUPERIOR. DAM Y DAW.*** | | |
| ***PROGRAMACIÓN, ENTORNOS DE DESARROLLO y BASES DE DATOS*** | | |
| ***Duración: 68 h*** | | ***Organización: Grupos de 3 personas (El profesor podrá efectuar los cambios que requiera)*** |

|  |
| --- |
| ***Reto 3***  ***TERMINAL DE AUTOBUSES: MÁQUINA EXPENDEDORA*** |
| ***Breve descripción***  En la empresa Termibus están satisfechos con nuestra labor en el segundo reto. Es tal el grado de satisfacción mostrado, que se han puesto en contacto con nuestro departamento de ventas y nos han encomendado la realización del software que gestione los billetes de los autobuses. |

|  |
| --- |
| **El Reto** |

**APLICACION VENTA DE BILLETES.**

Al igual que en el reto anterior, se nos ha pedido que creemos una aplicación con interfaces gráficas. De tal manera que todo la interacción del usuario con el sistema sea únicamente gráfica. Esto es algo que ya conocemos del reto anterior y por tanto, nuestro trabajo deberá de seguir la misma metodología de trabajo. Es decir, por un lado desarrollaremos las funcionalidades del sistema y por otro lado las funcionalidades del apartado gráfico. Donde no se aceptará, en modo alguno, encontrar código perteneciente al apartado gráfico en la parte de las funcionalidades y viceversa. A este respecto, se podría dar un paso más allá en nuestro paradigma de desarrollo, y interesarnos por el modelo-vista-controlador.

Más allá de la metodología de trabajo, los responsables de marketing nos han encomendado la siguientes tareas después de largas horas de negociaciones con la gente de Termibus:

El principal problema que tienen los de Termibus, es que aún viven entre la era analógica y digital. A este respecto, en su pliego de condiciones, se nos ha trasladado la necesidad de mejorar este sistema. La gente de marketing, que es experta en estas negociaciones, les ha vendido que una base de datos es lo mejor que se adapta a sus necesidades.

Para la realización de esta tarea los arquitectos de sistemas de Termibus se han comprometido a facilitarnos las tablas necesarias. Además se nos facilitarán los datos de conexión a la base de datos. Esta información la almacenaremos en un fichero de texto que deberemos leer.

Con esta información en la mano, nuestros compañeros de marketing, se han puesto en contacto con los arquitectos de nuestra área de desarrollo y se ha decidido que el siguiente diseño se podría adaptar a nuestras necesidades:

* Como queremos gestionar las distintas líneas y recorridos de Termibus trataremos la información de: el código de la línea y los autobuses (o el autobús) que recorren una línea y las poblaciones que componen esa línea.
* Con el objeto billete, gestionaremos el número de billete, la fecha o fechas, la línea o líneas del recorrido, la(s) parada(s) de origen y destino, el/los código de autobús(es), el precio del trayecto.
* Sobre el cliente guardaremos el dni, el nombre, sus apellidos, el sexo, fecha de nacimiento y su contraseña.
* La información referente a los autobuses será la siguiente: código de autobús, número de plazas de autobús, consumo/km y color.
* De las poblaciones, se guardará su nombre, el código postal y la relación de paradas que hay en dicha población.
* Por último, de las paradas guardaremos el código de la parada, la calle, el nombre de la parada y sus coordenadas geográficas.

Una vez que hemos entendido las restricciones impuestas a nuestro desarrollo desde el área de marketing en colaboración con los arquitectos de nuestro equipo de desarrollo, podemos ponernos a elaborar nuestra solución (los arquitectos de tu equipo de desarrollo serán tus profesores, cualquier cambio en el diseño de la aplicación deberá de tener el visto bueno del profesor al que le afecte dicho desarrollo).

Con todo esto en la mano, no debemos olvidar nuestro objetivo. Que no es otro que la gestión de la venta de billetes para las distintas líneas de autobuses.

A continuación, se nos ofrecen a **modo de ejemplo**, una serie de líneas gestionadas por nuestro cliente.

* Termibus-Plentzia
* Termibus-Muskiz
* Termibus-Balmaseda
* Termibus-Durango

FUNCIONAMIENTO DE LA APLICACION

En un primer momento se mostrará un mensaje de bienvenida y al interaccionar con la máquina la aplicación mostrará los diferentes trayectos o líneas existentes.

Una vez escogido uno de ellos se deberá escoger la parada origen y la parada destino, así como la fecha de realización del viaje Para ello se mostrará la relación de paradas del mismo, que aparecerán de la más cercana a la más lejana (para poder calcular el orden de dichas paradas podemos utilizar el dato de las coordenadas almacenado en el objeto parada. Mediante el cálculo de la distancia euclídea entre coordenadas, lograremos ordenarlas adecuadamente). Por ejemplo, para el trayecto Termibus-Mungia, un ejemplo de la relación de paradas podría ser el siguiente:

Línea: Bilbao-Mungia

1-Bilbao-Moyua

2-Bilbao-Univ.Deusto

3-Enekuri

4-Asua

5-Sondika

6-Derio

7-Artebakarra

8-Zabalondo

9-Urb La Bilbaina

10-Mungia

Al realizar dicha selección la aplicación nos mostrará, si hay plazas disponibles, el precio del billete. En el cálculo de plazas libres se reserva el trayecto completo sea cual sea el número de paradas que se recorran.

El precio del billete, vendrá dado por la distancia entre las dos paradas elegidas y el consumo del vehículo, la norma es que el beneficio por trayecto para la empresa es del 20% (Para este cálculo se supondrá que el autobús va lleno). Para calcular el precio del trayecto, solo nos falta conocer el precio de la gasolina. Por suerte para nosotros, a pesar de ser un valor que fluctúa, Termibus es un cliente premium para Petronor y le deja el diesel a un precio fijo de 80 céntimos por litro. En resumen, el precio del trayecto lo determinará la distancia, el precio de la gasolina, el consumo del vehículo y el número de asientos.

El billete que se reserve será válido para un solo cliente. Una vez finalizada esta reserva el sistema permitirá, antes de gestionar el cobro, realizar reservas de otros billetes, mostrando de forma clara la relación de billetes seleccionados con la información más relevante de los mismos.

Para la gestión del cobro, utilizaremos un sistema similar al del primer y segundo reto, donde la máquina siempre tratará de devolvernos el menor número posible de billetes y monedas.

En Termibus, están teniendo problemas con el robo de billetes y equipaje. Para el pillaje de pertenencias, han contratado a personal de seguridad y para el problema de los billetes han pensado en nosotros. De ahora en adelante, todos los billetes serán nominales y no se podrá montar en los autobuses sin acreditarse como es debido. Es por eso que, a la hora de comprar un billete, el documento de viaje deberá de acreditar correctamente los datos del usuario. Para ello, deberemos de desarrollar un sistema de login en el que te puedas dar de alta o simplemente introducir el dni y tu contraseña cuando estés registrado. Debido al aumento de descuideros en las inmediaciones de las estaciones, también se nos ha pedido que cuando se ingrese la contraseña, esta, aparezca con estrellas de tal manera que no sea legible.

Adicionalmente, se dará la posibilidad de guardar los ticket/factura de la compra en un fichero de texto, que incluirá los datos del recorrido, los datos del cliente, la fecha y hora de la compra/impresión. Además, toda compra de billetes deberá de quedar grabada en la base datos para poder elaborar un informe económico y eliminar aquellas líneas que resulten deficitarias.

A modo de corolario, el arquitecto del área de desarrollo de software, ha ido a un simposio sobre calidad en el desarrollo de software. Este viaje, le ha servido para venirse con un par de buenas prácticas que nos ha pedido que implementemos de ahora en adelante. Por un lado, los clientes que han utilizado la aplicación del segundo reto se han encontrado con numerosos bugs y se nos ha pedido que por favor, verifiquemos correctamente el funcionamiento del software antes de ponerlo en el mercado. Para ello, deberemos elaborar unos tests unitarios, por otro lado, se nos ha pedido que implementemos un sistema para la gestión de versiones y que además, permita el desarrollo en equipo. De tal manera, que cuando un compañero realice un avance en el desarrollo del producto, este, esté disponible para el resto de compañeros. La herramienta que nos ofrece estas funcionalidades, no es otra sino git.

Finalmente, como es costumbre en nuestra empresa, en la medida en la que sea posible, nos interesa utilizar software libre. Es por eso que para la gestión de la base de datos utilizaremos MySQL, que nos da soporte para lograr los niveles más altos de escalabilidad, seguridad, confiabilidad y tiempo de ejecución.

|  |
| --- |
| **Objetivos / Resultados de Aprendizaje** |

TÉCNICOS

1. *PROG- RA4.* Desarrolla programas organizados en clases, analizando y aplicando los principios de la programación orientada a objetos.
2. *PROG- RA5*. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
3. *PROG- RA6*. Escribe programas que manipulan información, seleccionando y utilizando tipos avanzados de datos.
4. *PROG- RA9*. Gestiona información almacenada en bases de datos relacionales, manteniendo la integridad y consistencia de los datos.
5. *EEDD-RA3.* Verifica el funcionamiento de programas diseñando y realizando pruebas.
6. *EEDD-RA4.* Optimiza código, empleando las herramientas disponibles en el entorno de desarrollo*.*
7. *EEDD-RA5.* Genera diagramas de clases, valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.
8. *BBDD-RA3*. Consulta la información almacenada en una base de datos, empleando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.
9. *BBDD-RA4.* Modifica la información almacenada en la base de datos, utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.
10. *BBDD-RA6.* 6. Diseña modelos relacionales normalizados, interpretando diagramas entidad/relación.

TRANSVERSALES

1. Eficacia y eficiencia en el trabajo.
2. Trabajo en equipo.
3. Comunicación.
4. Toma de decisiones.
5. Planificación y organización.

|  |
| --- |
| 1. **Tareas a realizar** |

La tarea a realizar debe contener los siguientes elementos:

* Generación de una tabla con las pruebas efectuadas, un manual de usuario acompañado de imágenes y diagrama de clases.
* Desarrollar una aplicación que cumpla los pliegos de condiciones expuestos en el reto 3.
* Desarrollar las clases necesarias para realizar las pruebas unitarias de los métodos creados.
* Preparar una presentación en la que tomen parte todos los miembros del grupo. La presentación no podrá exceder los 10 minutos bajo ninguna circunstancia. En dicha presentación se ejecutará el programa desarrollado en directo y se realizarán las pruebas pertinentes para demostrar su correcto funcionamiento. Su objetivo será la venta del programa a los representantes de Termibús.
* Planificar las tareas a desarrollar en cada dia y asignarlas a los distintos miembros del equipo. Estimar el tiempo dedicado a cada una de las tareas y recalcular los tiempos en los casos pertinentes.
* Analizar la base de datos suministrada y obtener el modelo entidad relación correspondiente.

|  |
| --- |
| 1. **Criterios de evaluación** |

* Competencias Técnicas % 70
* Competencias Transversales % 30

- 70% Grupales: 10% profesores, 40% grupo, 20% autoevaluación.

- 30% Presentación: 30% profesores.

* Si existiera una diferencia de 2 puntos o mayor entre la nota resultante la coevaluación+autoevaluación y de la resultante de la heteroevaluación, la parte de las grupales se calificará con la calificación más baja entre ellas.

|  |
| --- |
| 1. **Temporalización** |

68 horas (8+3+6)\*4

|  |
| --- |
| PRESENTACIÓN (2 horas) |

* Contrato
* Enunciado
* Rúbrica

|  |
| --- |
| PROTOTIPO IND (3 horas) |

* Auditoría (hora 5)
  + Prototipo individual
  + UML individual

|  |
| --- |
| PLANIFICACIÓN Y PROTOTIPO GRUPAL DIBUJADO  (6 horas) |

* Auditoría (hora 11)
  + Prototipo grupal
  + UML grupal
* Es necesario planificar el tiempo
  + To Do / Doing / Done
  + Trello (Google)

|  |
| --- |
| PROGRAMACION+BD+ENTORNOS (RUB) (15 horas) |

* Auditoría (hora 26)
  + Desarrollo

|  |
| --- |
| PROGRAMACION+BD+ENTORNOS (RUBRICA FUNCIONAMIENTO) (20 horas) |

* Auditoría (hora 46)
  + Desarrollo

|  |
| --- |
| DESARROLLO APLICACIÓN TERMINADA (20 horas) |

* Auditoría (hora 66)   **Nota parte técnica**
  + Desarrollo
  + Documentación
  + Presentación

|  |
| --- |
| PRESENTACION 2 horas |

* Evaluación (hora 68) + **Nota parte transversal**
  + Presentación