**Proyecto BiciUrban**

A picture containing logo

Description automatically generated

**Documento de Diseño**

EC-DD. Versión 3.0

30/04/2021

Estatus: Restringido

Jorge Blanco Rey

Ángel Casanova Bienzobas

Rodrigo Juez Hernández

Pablo Soëtard García

Grupo: 2391

Equipo: 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Versión | Fecha | Descripción del elemento / Versión | Realización | Validación | Revisión formal |
| EC-DC | 1.0 | 15/03/2021 | Diagrama de Clases del sistema inicial | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DC | 2.0 | 18/04/2021 | Diagrama de Clases modificado con la clase Experiencia | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DC | 3.0 | 30/04/2021 | Diagrama de Clases modificado con función generaInformesFacturacion(fecha:Date):FILE. | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS01 | 1.0 | 15/03/2021 | Diagrama Secuencia Autenticar Técnico | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS02 | 1.0 | 15/03/2021 | Diagrama Secuencia Añadir Nueva Bicicleta | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS19 | 1.0 | 15/03/2021 | Diagrama Secuencia Añadir Revisar Avería | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS13 | 1.0 | 15/03/2021 | Diagrama Secuencia Añadir Finalizar Alquiler | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS11 | 1.0 | 15/03/2021 | Diagrama Secuencia Alquilar Bicicleta con Abono | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS12 | 1.0 | 15/03/2021 | Diagrama Secuencia Denegar Servicio | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS4 | 1.0 | 18/04/2021 | Diagrama Secuencia Denegar Servicio | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DS5 | 1.0 | 30/04/2021 | Diagrama Secuencia Denegar Servicio | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DD | 1.0 | 15/03/2021 | Documento de Diseño inicial. | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DD | 2.0 | 18/04/2021 | Añadida una clase al diagrama de clases representando la Experiencia y una enumeración para la valoración. Modificado cliente. | Equipo de desarrollo | Equipo de Desarrollo |  |
| EC-DD  Línea Base | 3.0 | 30/04/2021 | Añadimos el Diagrama de Clases modificado, con su explicación en la sección 1 (Descripción de la arquitectura del Sistema) sobre las clases y métodos introducidos. Y modificamos la tabla de gestión de configuración. | Equipo de desarrollo | Equipo de Desarrollo | Equipo de Desarrollo/ Departamento de Calidad – Autoridad de Configuración |

**Resumen**

En este documento definimos como se debe implementar el sistema requerido, es decir, su diseño.

Inicialmente, encontramos la sección de descripción de la arquitectura del sistema, la sección 1. En ella, definimos de forma general las diferentes clases, atributos, métodos y relaciones que se deberán implementar. A través de este apartado nos podemos hacer una idea general del proyecto.

A continuación, en la sección 2, encontramos la representación gráfica de los descrito en el apartado anterior, el diagrama de clases. En este diagrama podemos encontrar de forma más detallada lo descrito anteriormente, así como los tipos de datos que debe tener cada atributo.

Finalmente, en la sección 3, encontramos los diagramas de secuencia de algunos casos de uso, que nos muestran cómo y a qué métodos se debe llamar en cada situación. Además, para cada diagrama encontramos una pequeña descripción del flujo de llamadas que se muestran para ese caso de uso.

**Índice de Contenidos**

1. Descripción de la Arquitectura del Sistema 5

2. Diagrama de Clases 7

3. Diagramas de Secuencia 8

3.1. Diagrama de Secuencia Caso 1 8

3.2. Diagrama de Secuencia Caso 2 8

3.3. Diagrama de Secuencia Caso 19 9

3.4. Diagrama de Secuencia Caso 13 10

3.5. Diagrama de Secuencia Caso 11 11

3.6. Diagrama de Secuencia Caso 12 12

3.7. Diagrama de Secuencia Caso 4 13

3.8. Diagrama de Secuencia Caso 5 13

4. Glosario 15

# 1. Descripción de la Arquitectura del Sistema

**Enumeraciones:**

Incluyen el estado de la bicicleta en ese momento, la modalidad de la experiencia y el tipo de bici al que se asocia un abono, este último es porque no se puede asociar con la clase de Bici en si ya que cada Bici será una instancia de esa clase.

**Clases:**

Las clases están divididas en los sistemas de Alquilar (Alquilar, Cliente, Abono, Experiencia), la interacción física con el sistema (Bicicleta, Estación, Ranura), y todo lo relacionado con el Técnico (Técnico, Incidencia, Informe, Estación).

**Aplicación** contiene la información de todos los clientes, estaciones, técnicos y averías, esto es para que pueda desempeñar las funciones de Informar y manipular el resto de las partes de la aplicación. Esta también generará informes para la dirección de la empresa, y se encargará de mandar los SMS en caso de que para ese tipo de cliente sea necesario y llamar a la correspondiente función de iniciaSesion.

**Bicicleta**: A parte de los campos habituales, incluimos un booleano “individual”, esto es porque cualquier tipo de bici (hibrida o clásica) puede ser familiar, además tiene una referencia a la ranura en la que se encuentra aparcada, si esta última es null significará que no está aparcada. La clase es abstracta puesto que no hay un tipo de bicicleta que no sea ni clásica ni híbrida, y cada tipo tiene su propia tarifa, condiciones de reparación y condiciones de mantenimiento.

**Ranura**: La ranura está contenida dentro de la estación, y tiene una referencia a la bici que está incluida en esa ranura, si esta referencia es null la ranura estará vacía. También incluye un campo para comprobar si está bloqueada.

**Estación**: Contiene todas las ranuras, gestiona su desbloqueo, registra las bicicletas nuevas añadidas al sistema, o reparadas, esto se hace en la estación y no en la aplicación porque cuando las registras la primera estación es la que usa el técnico.

También se encarga de recolectar todas las averías de las bicicletas para generar un informe. Finalmente, en caso de que se haya especificado que se deniega el servicio al usuario hay un método para enviar un error.

**Cliente**: Todos los clientes incluyen un número de teléfono y 0 o varias relaciones de alquileres, también tendrá incluido un método de pago. También tiene métodos para alquilar bicicletas que se encargan de asociarlas, métodos para informar en caso de avería, valorar una experiencia y el inicio de Sesión que se usará para confirmar que la identidad es correcta. El cliente a su vez puede ser pasajero, o abonado, las diferencias, aparte del método de inicio de sesión, las diferencias, son que el pasajero no tiene ninguna información asociada mientras que el abonado tiene un abono.

**Abono:** Este tiene una vigencia (medida en minutos restantes), cuantas bicicletas pueden estar alquiladas simultáneamente por ese abono y si ha sido penalizado (se almacena el tiempo de penalización).

**Alquiler**: Incluye la cantidad de bicis alquiladas, el tiempo de uso de ese alquiler (se almacena la longitud máxima cuando inicia para comprobar si hay un retraso), y si ha habido una incidencia, ya sea generado por el usuario o automática por un retraso.

**Incidencia**: Es una clase abstracta, ya que las incidencias solo pueden ser averías o retrasos. Esta clase contiene el atributo habitual, “id”, el identificador, y además un atributo “texto” de tipo char, donde irá escrito el motivo de la avería.

Luego, la clase Avería contiene el “coste”, que es un valor decimal indicando cuánto costaría la reparación de la avería, y la bicicleta a la que está relacionada la avería. Finalmente, la clase Retraso contiene el número entero de segundos que se ha retrasado la entrega de la bicicleta.

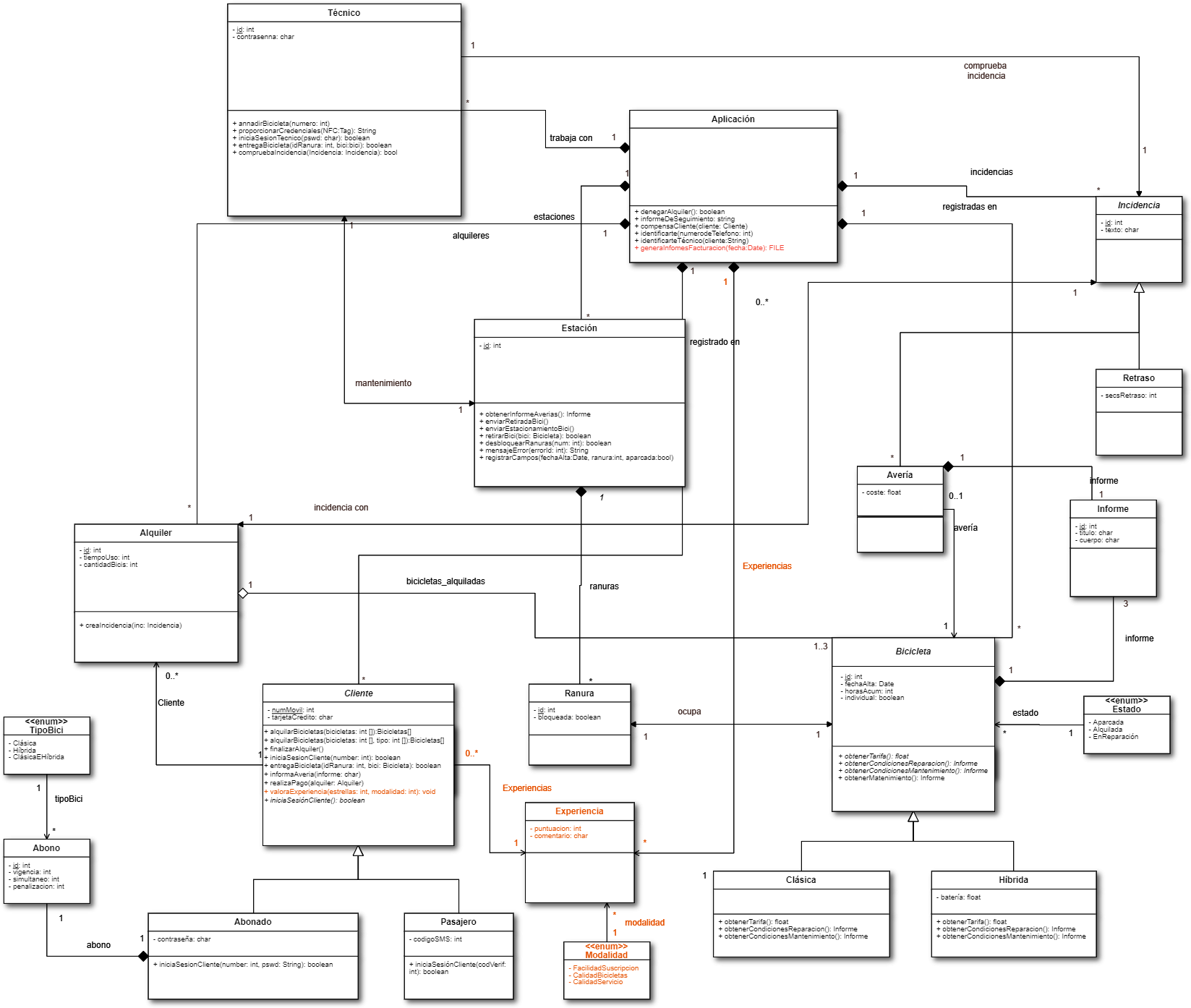
**Técnico**: La clase Técnico aparte del identificador, contiene “contrasenna” que es la contraseña con la que el técnico entrará en la aplicación, y la estación en la que está trabajando actualmente. También contiene los métodos annadirBicicleta, que se usa cuando el técnico añade una bicicleta en una estación; proporcionarCredenciales, usada cuando el técnico entra en el sistema con su etiqueta NFC; iniciaSesionTecnico; entregaBicicleta, usada para devolver una bicicleta tras haberla reparado; y compruebaIncidencia, que sirve para comprobar rápidamente si existen incidencias relacionadas con averías en la estación en la que el técnico está trabajando actualmente.

**Informe:** Esta clase tiene dos atributos, el título del informe y otro el cuerpo, este último es principalmente su contenido. El informe sirve principalmente que el técnico pueda obtener las bicicletas que necesitan reparación o mantenimiento en una estación.

**Experiencia:** Esta clase contiene dos atributos, puntuación y comentario. Esta clase sirve principalmente para ofrecer al usuario la posibilidad de valorar la facilidad de suscripción, calidad de las bicicletas o calidad de servicio según seleccione.

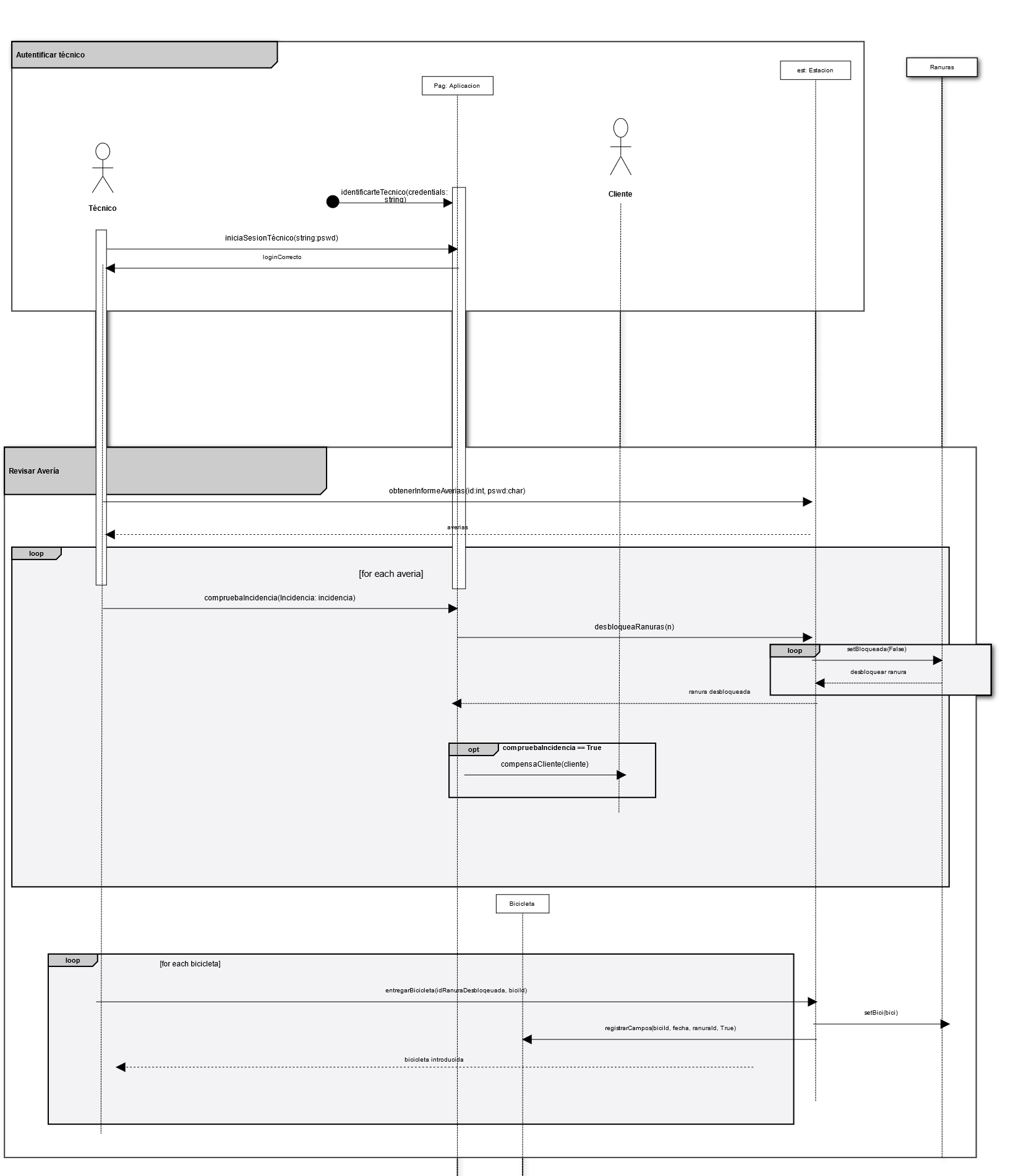
# 

# 2. Diagrama de Clases



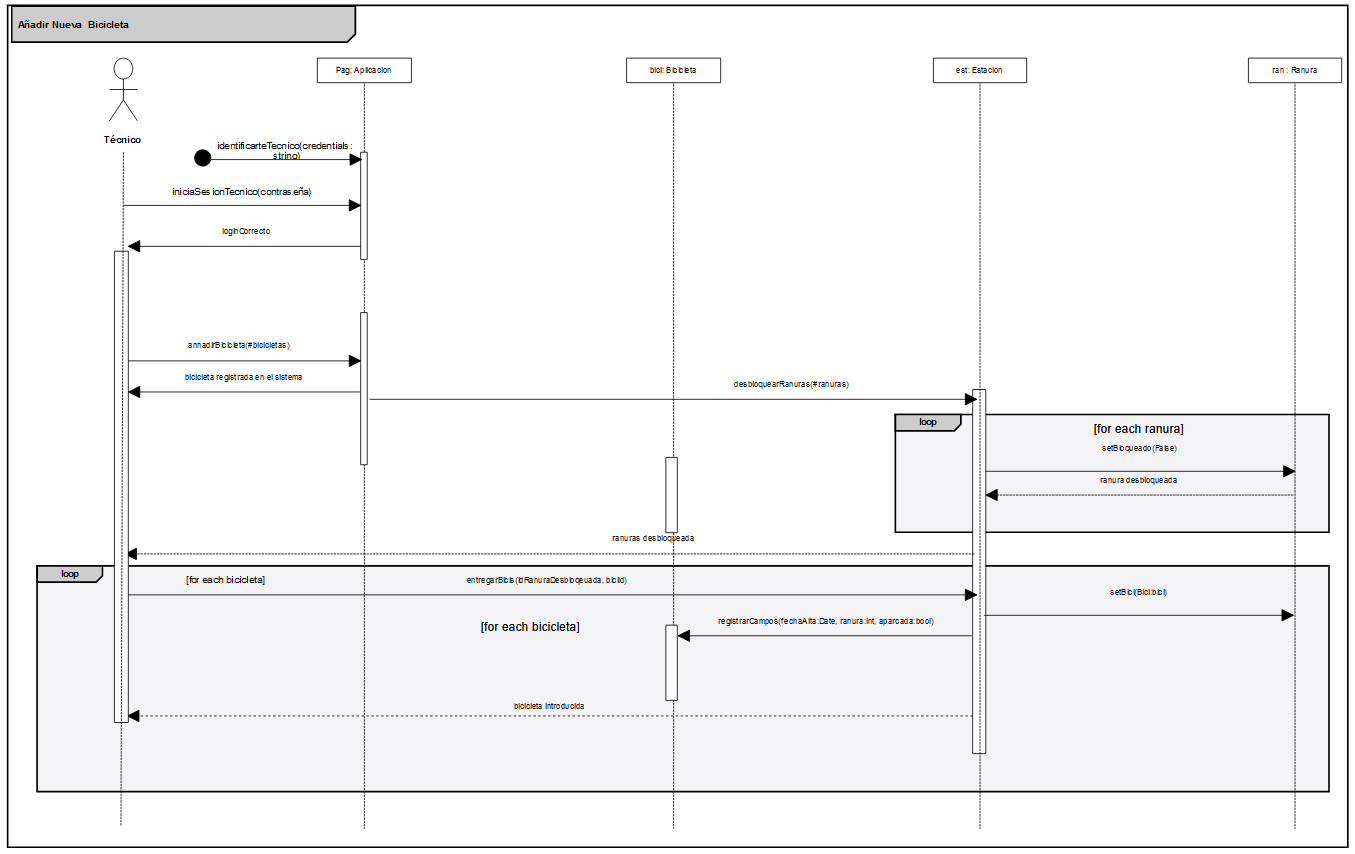
# 3. Diagramas de Secuencia

## 3.1. Diagrama de Secuencia Caso 1



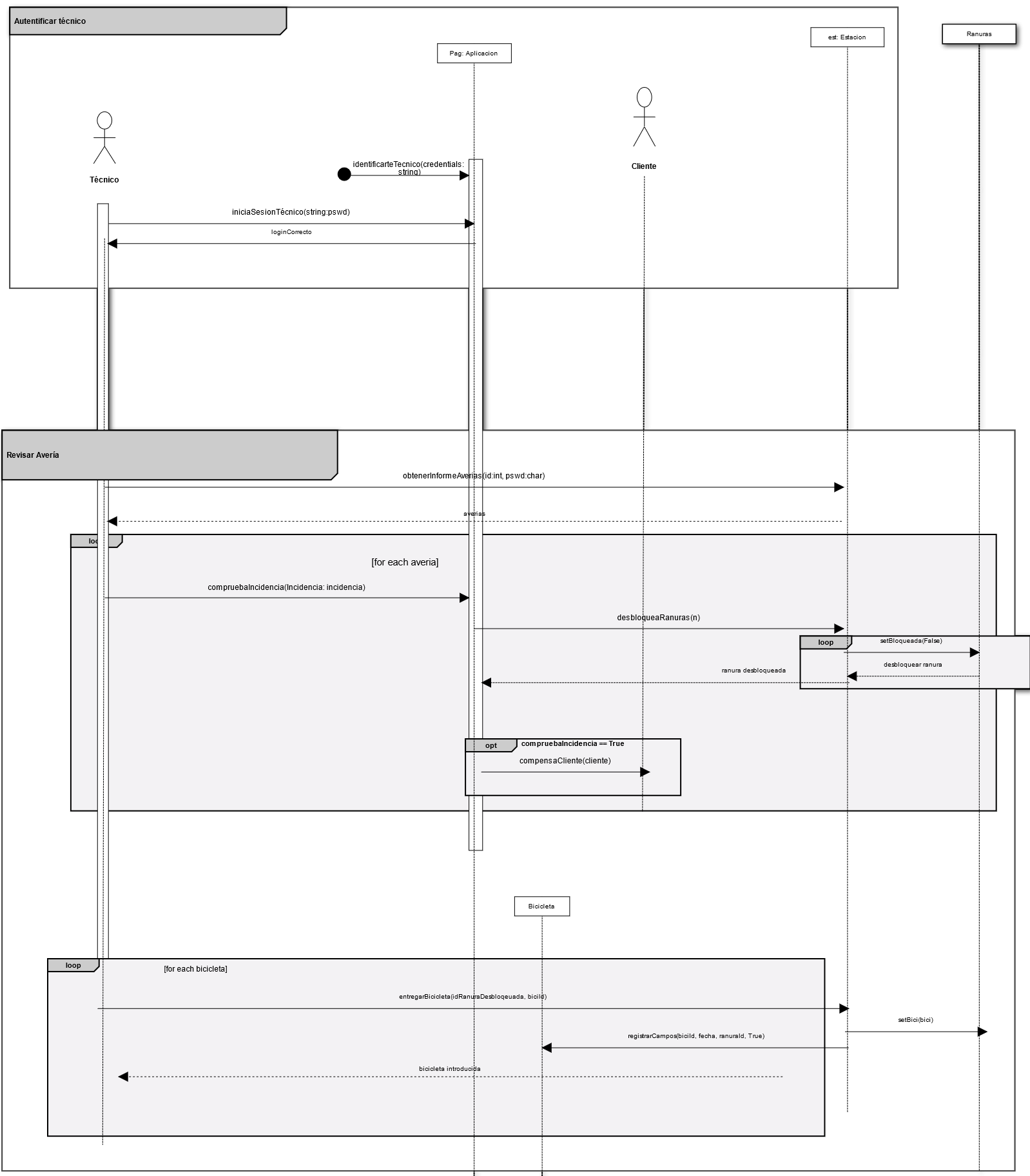
En este diagrama de secuencia se muestran los pasos que ha de seguir un técnico para identificarse. Como dispone de un dispositivo NFC, lo único que tiene que hacer es hacerse a un terminal y acercar su tarjeta, entonces la aplicación validará sus credenciales y el técnico habrá iniciado su sesión correctamente.

## 3.2. Diagrama de Secuencia Caso 2



En este diagrama de secuencia se muestran los pasos que ha de seguir un técnico para añadir una nueva bicicleta en el sistema. Tras haber iniciado sesión (como se explica en la sección 3.1). Tras esto, es el técnico indica en la terminal que quiere añadir una bicicleta y comienza un proceso iterativo en el cual la terminal va abriendo y cerrando las ranuras cuando el técnico introduce las bicicletas que desea añadir a esa terminal.

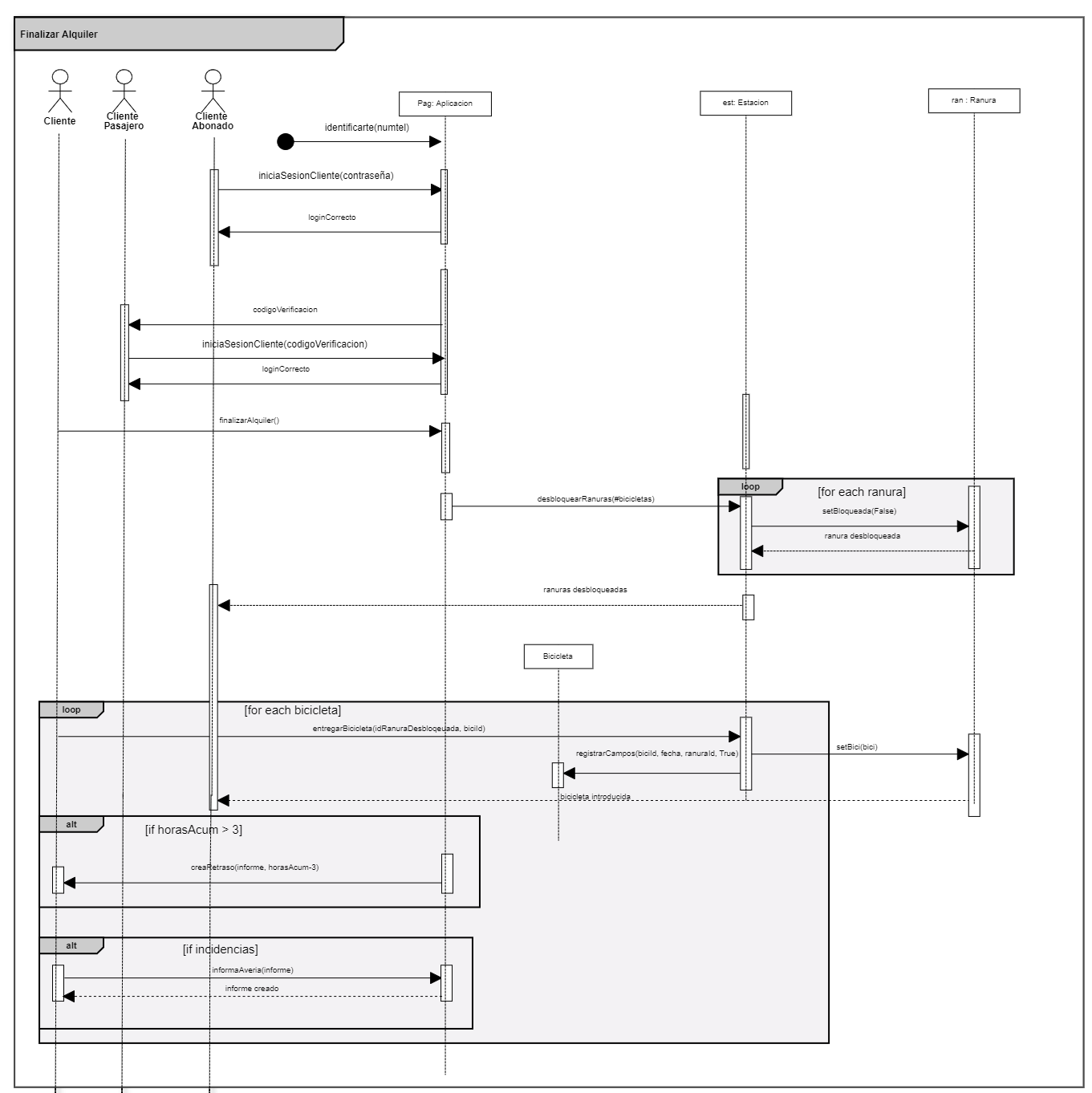
## **3.3. Diagrama de Secuencia Caso 19**



En este diagrama de secuencia se muestran los pasos que ha de seguir un técnico para revisar una avería en el sistema. Tras haber iniciado sesión (como se explica en la sección 3.1), el técnico podrá acceder a un menú en la terminal donde recibe los informes de las averías (si existen), en tal caso por cada avería, retira una bicicleta de la terminal para examinarla y tras realizar un examen técnico al vehículo decide si tiene o no tiene que compensar al cliente en cuestión.

En el caso pertinente, el técnico arreglará la bicicleta y tras ello la devolverá a la terminal una vez la bicicleta este en perfectas condiciones para poder circular.

## **3.4. Diagrama de Secuencia Caso 13**



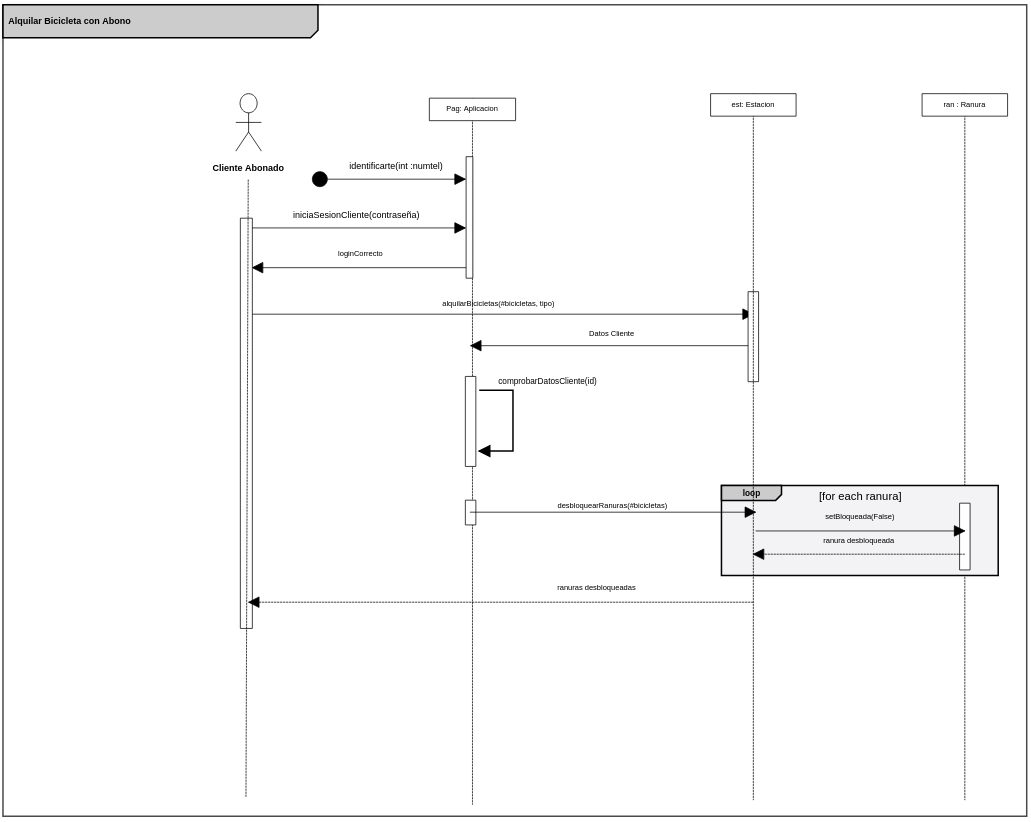
Este diagrama de secuencia representa el proceso de finalizar alquiler. Como vemos tenemos tres actores, cliente pasajero y abonado, que se identifican de forma independiente, ya que la forma en que cada uno de ellos entra al sistema es distinta como veremos a continuación. Luego, tenemos el actor cliente, que representa las acciones comunes que se realizan para los pasajeros y abonados.

Como vemos, ambos clientes se identifican en el sistema mediante su número de teléfono y a través del método identificarte, luego el abonado introduce su contraseña y el pasajero recibe un código de verificación que introduce en la aplicación.

A continuación, el cliente indica que desea finalizar el alquiler, y se llama al método finalizarAlquiler, por lo que, si se encuentra en una estación con ranuras disponibles, se desbloquean mediante el método desbloquearRanuras y el bucle que podemos observar, si no existen ranuras disponibles en esa estación el proceso finaliza.

A partir de entonces, para cada bicicleta (por ello se encuentra dentro de un bucle) después de entregarla se pueden abrir dos tipos de incidencias, un retraso en la entrega, a través del método creaRetraso que es invocado por el sistema si el número de horas de uso de la bicicleta supera las tres horas. Y el otro tipo de incidencia la crea el usuario, por tanto, si quiere abrir una incidencia por una avería, llamamos al método informaAveria.

## **3.5. Diagrama de Secuencia Caso 11**

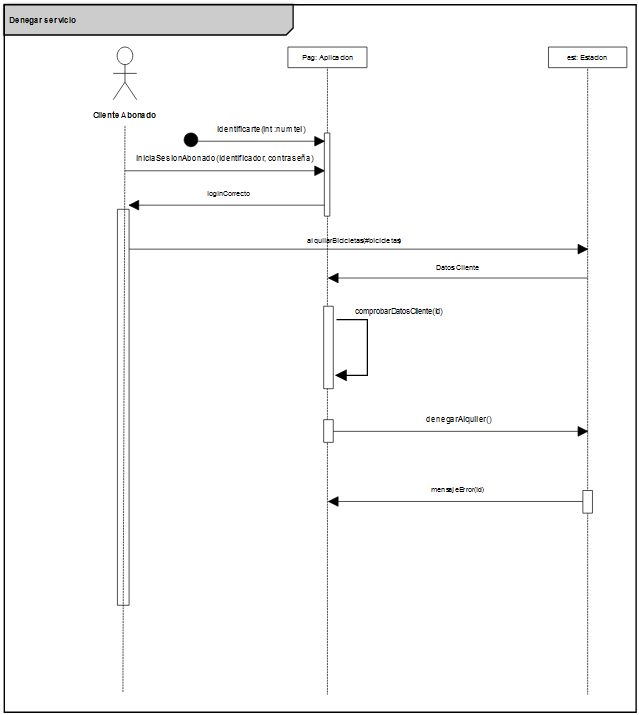


Este diagrama de secuencia representa el proceso que lleva a cabo un cliente abonado para alquilar una bicicleta. En este caso nuestro único actor es el Cliente Abonado, que lo primero que hace es identificarse en la aplicación con su número de teléfono a través del método identificarte, a continuación, el cliente abonado introduce su contraseña que es enviada a la Aplicación a través del método iniciaSesionCliente, tras ello la aplicación notifica loginCorrecto.

Una vez el cliente abonado se ha identificado correctamente, este comienza el alquiler proveyendo el número de bicicletas y el tipo que desea alquilar con el método alquilarBicicletas, la estación comprueba que tiene disponibles al menos el número de bicicletas del tipo que el cliente ha solicitado y devuelve algunos datos del cliente a la aplicación, estos datos son procesados por la aplicación para comprobar que al cliente no se le ha denegado el servicio por mal uso del mismo, que su abono no ha caducado, que con el abono que tiene puede solicitar el número y tipo de bicicletas que ha pedido y que aún no ha superado el número de alquileres del día, todo ello mediante el método comprobarDatosCliente.

Tras haber hecho las comprobaciones oportunas, se pasa a desbloquear el número de ranuras de la estación que ha solicitado el cliente con el método desbloquear ranuras, la estación entra en un loop para desbloquear todas las ranuras y confirma al cliente que todas las bicicletas que ha solicitado se han desbloqueado correctamente.

## **3.6. Diagrama de Secuencia Caso 12**

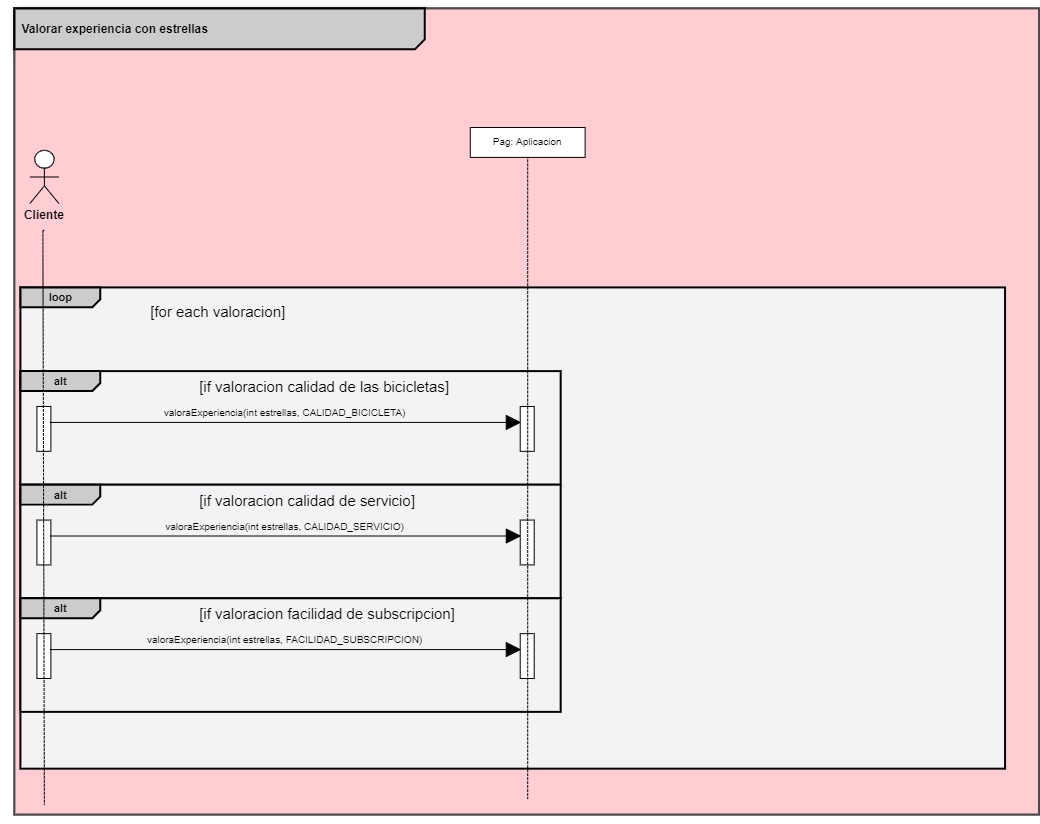


En este diagrama de secuencia, se muestra el proceso mediante el cual el sistema, rechaza la petición del alquiler.

Si un usuario ha tenido conductas inapropiadas como devolver las bicicletas tarde, no cuidar bien cualquier material o por cualquier razón que el staff considere oportuna, un usuario puede perder el derecho a realizar alquileres temporalmente o de manera indefinida, a continuación, explicamos el proceso de denegación de servicio del alquiler.

Cuando el usuario, inicia sesión en terminal, puede seguir viendo sus tiques, así como cualquier otra información vinculada a el mismo. Pero cuando el usuario intenta alquilar una bicicleta, la terminal manda un identificador de cliente al servidor central de la aplicación, el cual, mediante un método, decide si el usuario puede realizar el alquiler o no, después, este servidor devuelve la respuesta al terminal, donde el usuario recibe un mensaje de error, indicándole que no puede realizar el alquiler y además se le informa de la razón por la cual no se le dará servicio.

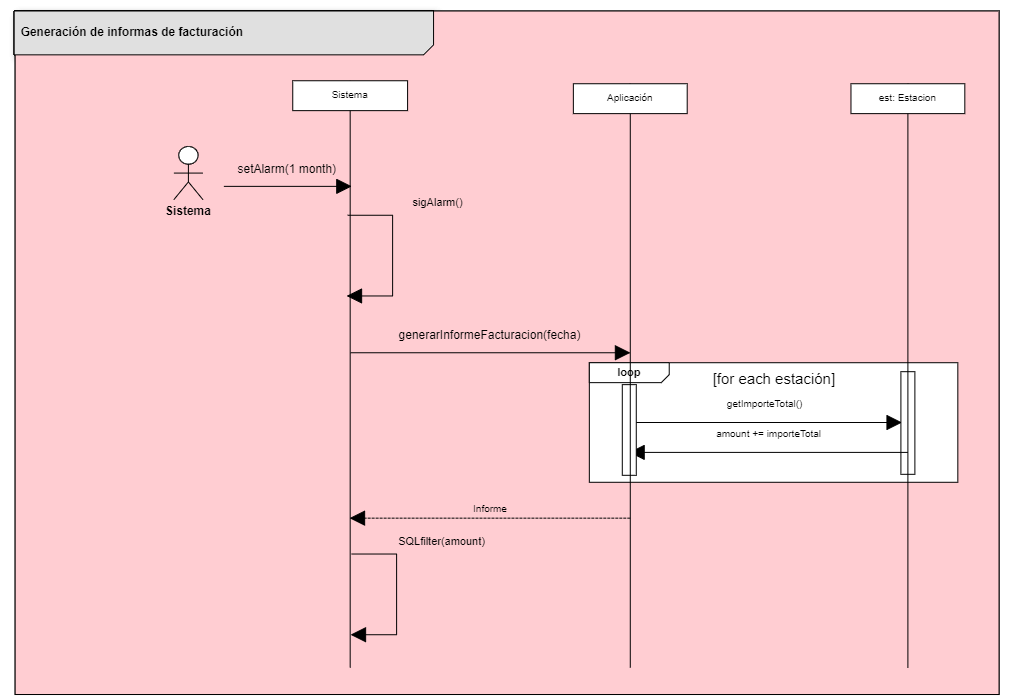
## **3.7. Diagrama de Secuencia Caso 4**



En este diagrama de secuencia se muestrea el proceso mediante el cual un cliente abonado o no abonado podra valorar la experiacia que ha tenido al utilizar nuestra aplicación mediante la asignacion de 5 estrellas siendo 1 estralla la mas baja y 5 la mas alta.

Para desarrollar esta funcion, hemos codificado la funcion "valorarExperiencia(int estrellas, int tipo)", donde tipo es la clase de esperiencia que se quiere valorar de entre las tres disponibles (calidad de las bicicletas, calidad de servicio y facilidad de subcripcion).

## **3.8. Diagrama de Secuencia Caso 5**



El sistema pre-programa un proceso través de una señal del sistema SIGALRM que se ejecuta una vez al mes, llamado a un método estático de la clase Aplicación, el cual cogerá el importe total de cada terminal a lo largo del mes, no será necesario añadir más clases ni funciones a los terminales ya que todos los datos están registrados en la base de datos del sistema y se pueden filtrar a través de una consulta SQL.

# 4. Glosario

|  |  |
| --- | --- |
| Término | Descripción |
| **Pasajero** | Cliente que no tiene ningún abono en curso. |
| **Abondado** | Cliente que tiene un abono en curso. En este se especifica el periodo de vigencia desde su compra, tipo de bicicletas que puede alquilar y el número de ellas que puede alquilar de forma simultánea. |
| **Técnico** | Empleado que se encarga de la reparación y mantenimiento de las bicicletas. |