**GESTION DE DATOS 2019**

* **Cuatrimestre.** 1ro
* **Curso.** K3014.

**TRABAJO PRÁCTICO**

**CRUCEROS FRBA**

|  |  |
| --- | --- |
| **GRUPO TIRANDO\_QUERIES (#24)** | |
| **MIEMBRO** | **LEGAJO** |
| LOPEZ SURRA, Rodrigo | 152.725-3 |
| CODA, Agustín | 155.974-6 |
| ZITO, Alejo | 156.626-9 |
| MOSSE, Lautaro | 153.902-4 |

ÍNDICE

[MODELADO DE DATOS 3](#_Toc12297521)

[DIAGRAMA DE ENTIDAD RELACION 3](#_Toc12297522)

[TABLA USUARIO 3](#_Toc12297523)

[TABLA ROL 4](#_Toc12297524)

[TABLA PERMISO 4](#_Toc12297525)

[TABLA ROL\_USUARIO 4](#_Toc12297526)

[TABLA PERMISO\_ROL 5](#_Toc12297527)

[TABLA CLIENTE 5](#_Toc12297528)

[TABLA PASAJE 5](#_Toc12297529)

[TABLA ESTADO\_PASAJE 6](#_Toc12297530)

[TABLA PAGO 6](#_Toc12297531)

[TABLA RESERVA 7](#_Toc12297532)

[TABLA ESTADO\_RESERVA 7](#_Toc12297533)

[TABLA RUTA\_VIAJE 7](#_Toc12297534)

[TABLA RECORRIDO 8](#_Toc12297535)

[TABLA PUERTO 8](#_Toc12297536)

[TABLA TRAMO 8](#_Toc12297537)

[TABLA CRUCERO 8](#_Toc12297538)

[TABLA FABRICANTE 9](#_Toc12297539)

[TABLA MODELO\_CRUCERO 9](#_Toc12297540)

[TABLA MANTENIMIENTO 9](#_Toc12297541)

[TABLA CABINA 9](#_Toc12297542)

[TABLA TIPO\_CABINA 9](#_Toc12297543)

[DECISIONES DE DISEÑO E IMPLEMENTACIONES 10](#_Toc12297544)

[ARQUITECTURA 10](#_Toc12297545)

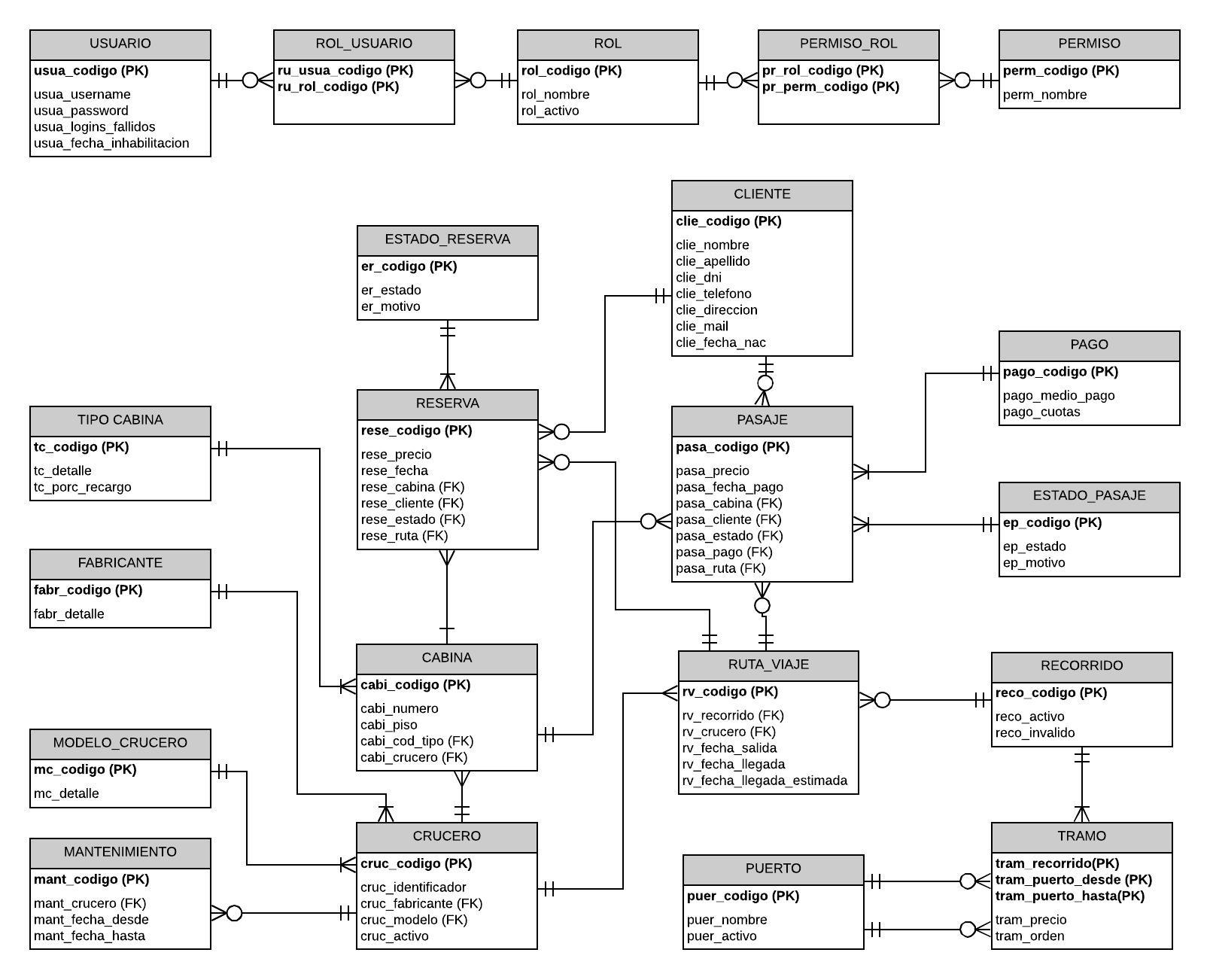
[PROCESO DE LOGIN 10](#_Toc12297546)

[ASPECTOS VARIOS DE INTERPRETACIÓN DEL ENUNCIADO 11](#_Toc12297547)

[PANTALLAS DE LA APLICACION 13](#_Toc12297548)

# MODELADO DE DATOS

## DIAGRAMA DE ENTIDAD RELACION

****

## TABLA USUARIO

Almacena información del usuario.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **usua\_codigo (PK)** | Código del usuario |
| **usua\_username** | Nombre de usuario. Debe ser único. |
| **usua\_password** | Contraseña del usuario. Almacenada con cifrado SHA256. |
| **usua\_logins\_fallidos** | Cantidad de logins fallidos del usuario |
| **usua\_fecha\_inhabilitacion** | Fecha en que se inhabilita un usuario |

ACLARACIONES

* Se definió el set de usuarios:

|  |  |
| --- | --- |
| **USERNAME** | **PASSWORD** |
| admin | w23e |
| admin2 | w23e |

## TABLA ROL

Almacena información de los roles que permiten al usuario acceder a las distintas funcionalidades de la APP.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **rol\_codigo (PK)** | Código del rol |
| **rol\_nombre** | Nombre del rol. Debe ser único |
| **rol\_activo** | Flag que permite identificar la baja lógica de un rol |

ACLARACIONES

* Se definieron los roles: Administrativo y Cliente.

## TABLA PERMISO

Almacena información de los permisos que permiten al rol acceder a las distintas funcionalidades de la APP.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **perm\_codig (PK)** | Código del permiso |
| **perm\_nombre** | Nombre del permiso. Debe ser único |

ACLARACIONES

* Se definieron los permisos: ABM\_ROL, ABM\_PUERTO, ABM\_RECORRIDO, ABM\_CRUCERO, GENERAR\_VIAJE, COMPRAR\_PASAJE, RESERVA\_PASAJE, PAGO\_RESERVA, LISTADO\_ESTADISTICO.

## TABLA ROL\_USUARIO

Almacena la relación entre los usuarios y sus roles asociados.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **ru\_usua\_codigo (PK)** | Código del usuario |
| **ru\_rol\_codigo (PK)** | Código del rol asociado |

ACLARACIONES

* Se definieron las siguientes relaciones:

|  |  |
| --- | --- |
| **USUARIO** | **ROL** |
| admin | Administrativo |
| admin2 | Administrativo |

## TABLA PERMISO\_ROL

Almacena la relación entre los roles y sus permisos asociados.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **pr\_rol\_codigo (PK)** | Código del rol |
| **pr\_perm\_codigo (PK)** | Código del permiso asociado |

ACLARACIONES

* Se definieron las siguientes relaciones:

|  |  |
| --- | --- |
| **ROL** | **PERMISO** |
| Administrativo | TODOS |
| Cliente | GESTIONAR\_PASAJE, PAGO\_RESERVA |

## TABLA CLIENTE

Almacena información de los clientes.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **clie\_codigo (PK)** | Código del cliente |
| **clie\_nombre** | Nombre del cliente |
| **clie\_apellido** | Apellido del cliente |
| **clie\_dni** | Número de documento del cliente |
| **clie\_telefono** | Teléfono del cliente |
| **clie\_direccion** | Dirección del cliente |
| **clie\_mail** | Mail del cliente |
| **clie\_fecha\_nac** | Fecha de nacimiento del cliente |
| **clie\_duplicado** | Permite identificar si existe un cliente con DNI duplicado. |

ACLARACIONES

* El valor del campo **clie\_duplicado** identifica a clientes con DNI duplicados dado que interpretamos el enunciado establece que la aplicación define la univocidad a nivel DNI (regla de negocio). Esto lo vemos, por ejemplo, al momento de la compra de un pasaje: se solicita se completen todos los datos del cliente (si existe en la base de datos) cuando el mismo hace ingreso de su DNI.

## TABLA PASAJE

Almacena información de los pasajes comprados.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **pasa\_codigo (PK)** | Código del pasaje |
| **pasa\_precio** | Precio del pasaje |
| **pasa\_fecha\_pago** | Fecha de compra/pago del pasaje |
| **pasa\_cabina (FK)** | Código de la cabina al que está asociada |
| **pasa\_cliente (FK)** | Código del cliente al que está asociado |
| **pasa\_estado (FK)** | Código del estado al que está asociado |
| **pasa\_pago (FK)** | Código del pago al que está asociado |
| **pasa\_ruta (FK)** | Código de la ruta de viaje al que está asociada |

## TABLA ESTADO\_PASAJE

Almacena los distintos estados que un pasaje puede tener.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **ep\_codigo (PK)** | Código del estado del pasaje |
| **ep\_estado** | Estado del pasaje |
| **ep\_motivo** | Motivo del estado del pasaje |

ACLARACIONES

* Se definieron los siguientes estados:

|  |  |
| --- | --- |
| **ESTADO** | **MOTIVO** |
| Vigente | Pasaje vigente |
| Desconocido | Sin información |
| Cancelado | Desperfecto técnico en crucero |
| Cancelado | Crucero completo VU |
| Cancelado | Baja de recorrido |

* El **estado desconocido** se asocia a los datos migrados, dado que no se tiene la información necesaria para garantizar que hayan sido o no cancelados.

## TABLA PAGO

Almacena los distintos tipos de pagos existentes y sus correspondientes cuotas.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **pago\_codigo (PK)** | Código del pago |
| **pago\_medio\_pago** | Medio de pago |
| **pago\_cuotas** | Cantidad de cuotas para el medio de pago |

ACLARACIONES

* Se definió que los pagos se registran colocando una fecha en la tabla de pasaje y referenciando el medio del mismo a esta tabla. La misma limita todos los medios de pago, junto con sus cuotas disponibles. Esto se hizo principalmente para evitar generar datos adicionales sobre entidades poco relevantes al trabajo práctico, y que son brevemente mencionadas en el enunciado.
* Se definieron los siguientes medios de pago:

|  |  |
| --- | --- |
| **MEDIO** | **CUOTAS** |
| Efectivo | 0 |
| TD Santander | 0 |
| TC Santander | 3 |
| TC Ciudad | 6 |
| Desconocido | NULL |

* El **medio desconocido** se asocia a los datos migrados, dado que no se tiene la información necesaria para garantizar cómo se realizó el pago.

## TABLA RESERVA

Almacena información de los pasajes reservados.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **rese\_codigo (PK)** | Código de la reserva |
| **rese\_precio** | Precio de la reserva |
| **rese\_fecha** | Fecha de la reserva |
| **rese\_cabina (FK)** | Código de la cabina al que está asociada |
| **rese\_cliente (FK)** | Código del cliente al que está asociada |
| **rese\_estado (FK)** | Código del estado de la reserva al que está asociada |
| **rese\_ruta (FK)** | Código de la ruta de viaje al que está asociada |

## TABLA ESTADO\_RESERVA

Almacena los distintos estados que una reserva puede tener.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **er\_codigo (PK)** | Código del estado de la reserva |
| **er\_estado** | Estado de la reserva |
| **er\_motivo** | Motivo del estado de la reserva |

ACLARACIONES

* Se definieron los siguientes estados:

|  |  |
| --- | --- |
| **ESTADO** | **MOTIVO** |
| Vigente | Reserva vigente |
| Vencida | Supero la cantidad de días permitidos |
| Pagado | Se realizó el pago |
| Cancelado | Cliente desiste |
| Desconocido | Sin información |

* El **estado desconocido** se asocia a los datos migrados, dado que no se tiene la información necesaria para garantizar que hayan sido canceladas o vencidas.

## TABLA RUTA\_VIAJE

Almacena información de las rutas de viajes ofrecidas.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **rv\_codigo (PK)** | Código de la ruta de viaje |
| **rv\_recorrido (FK)** | Código del recorrido al que está asociado |
| **rv\_crucero (FK)** | Código del crucero al que está asociado |
| **rv\_fecha\_salida** | Fecha de salida del crucero |
| **rv\_fecha\_llegada** | Fecha de llegada real del crucero |
| **rv\_fecha\_llegada\_estimada** | Fecha de llegada estimada del crucero |
|  |  |

## TABLA RECORRIDO

Almacena los distintos recorridos existentes.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **reco\_codigo (PK)** | Código del recorrido |
| **reco\_activo** | Flag que permite identificar la baja lógica de un recorrido |
| **reco\_invalido** | Flag que permite identificar la incerteza de un registro |

ACLARACIONES

* Ante la decisión de que cada tramo de la tabla Maestra se considere como un recorrido de un solo tramo (comunicado en el grupo de mail), definimos el flag **reco\_invalido** para aquellos recorridos distintos que comparten el recorrido id, dado que no pueden identificarse unívocamente.

## TABLA PUERTO

Almacena los distintos puertos existentes.

|  |  |  |
| --- | --- | --- |
| **CAMPO** | **DESCRIPCIÓN** | |
| **puer\_codigo (PK)** | | Código de un puerto |
| **puer\_nombre** | | Nombre de un puerto. Debe ser único. |
| **puer\_activo** | | Flag que permite identificar la baja lógica de un puerto |

## TABLA TRAMO

Almacena los distintos recorridos existentes.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **tram\_recorrido (PK)** | Código del recorrido |
| **tram\_puerto\_desde (PK)** | Código del puerto origen |
| **tram\_puerto\_hasta (PK)** | Código del puerto destino |
| **tram\_precio** | Precio del tramo |
| **tram\_orden** | Número de orden del tramo en un recorrido |

ACLARACIONES

* El campo **tram\_orden** se definió con el objetivo de poder identificar la secuencia de los tramos de un recorrido. Esto es porque en los casos de edición de los recorridos resulta más práctico y más usable (user friendly) de cara al administrador.

## TABLA CRUCERO

Almacena información de los cruceros existentes.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **cruc\_codigo (PK)** | Código del crucero |
| **cruc\_identificador** | Identificador del crucero |
| **cruc\_fabricante (FK)** | Código del fabricante del crucero al que está asociado |
| **cruc\_modelo (FK)** | Código del modelo del crucero al que está asociado |
| **cruc\_activo** | Flag que permite identificar la baja lógica de un crucero |

## TABLA FABRICANTE

Almacena información de los distintos fabricantes de cruceros existentes.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **fabr\_codigo (PK)** | Código del fabricante |
| **fabr\_detalle** | Nombre del fabricante |

## TABLA MODELO\_CRUCERO

Almacena información de los distintos modelos de cruceros existentes.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **mc\_codigo (PK)** | Código del modelo |
| **mc\_detalle** | Nombre del modelo |

## TABLA MANTENIMIENTO

Almacena información de los mantenimientos de cruceros.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **mant\_codigo (PK)** | Código de mantenimiento del crucero |
| **mant\_crucero (FK)** | Código del crucero al que está asociado |
| **mant\_fecha\_desde** | Fecha de ingreso en mantenimiento |
| **mant\_fecha\_hasta** | Fecha de egreso de mantenimiento |

## TABLA CABINA

Almacena información de las cabinas de los cruceros.

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN** |
| **cabi\_codigo (PK)** | Código de la cabina |
| **cabi\_numero** | Número de la cabina |
| **cabi\_piso** | Piso de la cabina |
| **cabi\_cod\_tipo (FK)** | Código del tipo de cabina |
| **cabi\_crucero (FK)** | Código del crucero al que pertenece |

## TABLA TIPO\_CABINA

Almacena los distintos puertos existentes.

|  |  |  |
| --- | --- | --- |
| **CAMPO** | **DESCRIPCIÓN** | |
| **tc\_codigo (PK)** | | Código del tipo de una cabina |
| **tc\_detalle** | | Nombre del tipo de cabina |
| **tc\_porc\_recargo** | | Porcentaje de recargo del precio |

# DECISIONES DE DISEÑO E IMPLEMENTACIONES

## ARQUITECTURA

La solución se definió en 3 capas: una de presentación, una capa de negocio y, por último, la de acceso a datos. Esta distribución es útil para organizar el código según responsabilidades, y permitiría sustituir fácilmente componentes de ser necesario.

Además, la aplicación desktop se diseñó en base a una arquitectura MVVM de 3 capas: model (domain objects), Views (forms) y ViewModels:

* **MODELS**. Son la representación en POO de las tablas del proyecto. Las clases finalizadas en “DAO” son las encargadas de ejecutar las consultas SQL y retornar las entidades del dominio.
* **VIEWS.** Son los formularios de Windows Forms. Contienen diversos controles como botones o inputs de texto, y se encargan de comunicar los eventos generados por el usuario al operar la UI.   
  Estas clases están encargadas de conectar los diversos componentes visuales con su correspondiente ViewModel. Este proceso se realiza mediante el Binding, donde cada control visual se enlaza a una propiedad del ViewModel y, de esta manera, el objeto está siempre actualizado con la información de la UI.   
  En algunos casos, fue necesario realizar Two-Way Binding ya que el ViewModel debía actualizar valores de la vista, como por ejemplo cuándo el usuario ingresa el DNI y, en consecuencia, se debe buscar en la db los datos de Cliente asociados a dicho DNI y autocompletar los campos de texto.
* **VIEW-MODELS.** Son el enlace entre las vistas y los objetos del dominio. Contienen el código correspondiente a la lógica del negocio, como las validaciones que se ejecutan al crear una entidad. Por ejemplo, el cálculo del monto total a medida que el usuario va seleccionando cabinas al comprar un pasaje.  
  Cada formulario, tiene asociado su correspondiente ViewModel de manera de garantizar esta separación de responsabilidades.

En cuanto a la segurización de la aplicación, al ser de escritorio, se definió guardar al usuario logueado en memoria. Dicha clase contiene información del usuario y un listado con todos sus permisos, sobre los que se define cuáles botones de la interfaz visual se muestran y cuáles se ocultan. De esta manera, quedan definidos los accesos a cada sección de la aplicación en base a los permisos del rol del usuario. Si existiera interés en agregar nuevas vistas y segurizarlas, sería cuestión de agregar los permisos a la base de datos y codificar las reglas acordes a estas nuevas vistas.

## PROCESO DE LOGIN

El proceso de login se implementa en el stored procedure “SP\_AUTENTICAR\_USUARIO”.

1. Se realiza una validación que el usuario existe y no esté bloqueado. Para este último punto, se verifica si la cantidad de intentos fallidos es mayor o igual a 3.
2. Si la cantidad de intento fallidos es mayor o igual a 3, el usuario quedará bloqueado durante 10 minutos.
3. Se comparan las contraseñas hasheadas.

## ASPECTOS VARIOS DE INTERPRETACIÓN DEL ENUNCIADO

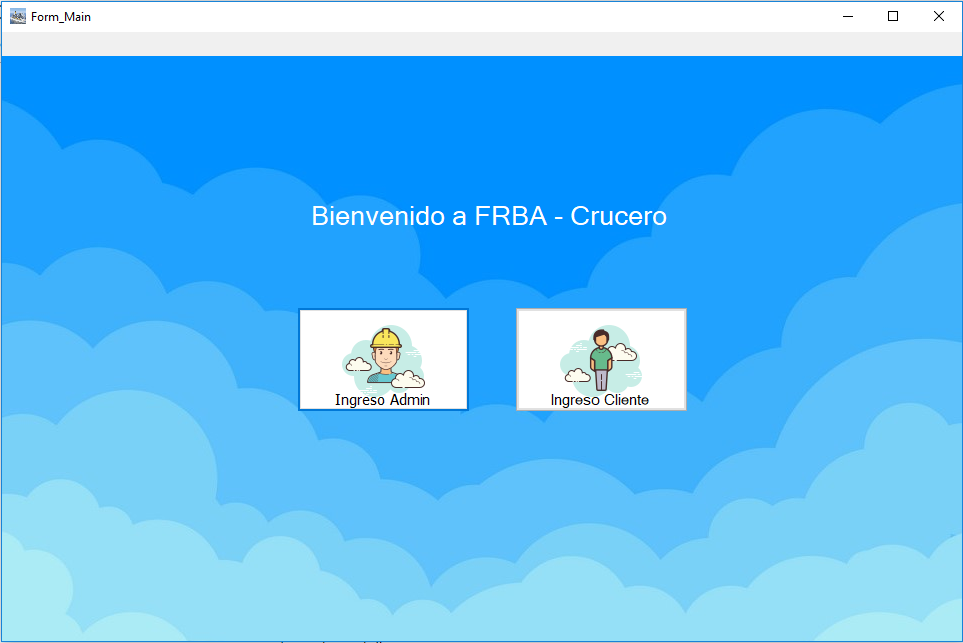
* **Conocimiento entre un pasaje y una reserva**. El enunciado no especifica que la reserva y el pasaje necesiten vincularse, dado que no hay ninguna funcionalidad que se valga de esta información. Si fuese necesario, se debiera agregar una FK en la tabla de pasajes, que se complete al momento de pagar una reserva.
* **Edición de cruceros**. Se definió que la edición de cruceros no permite agregar o quitar cabinas dado que resulta ilógico en un escenario real. Es decir, un crucero real no tiene agregados o quitados de cabinas. En tal caso, si una cabina se vuelve no utilizable, se debiera realizar una baja lógica de la misma pero esta situación implica definir una regla de negocio para entender cómo tratar con los pasajes vendidos y la misma no está definida en el enunciado.
* **Reemplazo de cruceros**. Favoreciendo la simplicidad, al enviar un crucero a mantenimiento y optar por su reemplazo, se listan todos los cruceros con suficientes cabinas disponibles para el reemplazo, sin importar los tipos de cabina. Es decir, al reemplazar el crucero pueden reasignarse cabinas con diferente tipo al contratado. Consideramos que el enunciado no explicita una estrategia de reemplazo en especial y que, al tratarse de un caso atípico, el negocio podría permitir dicha regla. Se suma a esto la dificultad para obtener la totalidad de las cabinas por crucero, ya que no contamos con dicha información de a partir de la migración (solamente tenemos las cabinas de aquellos pasajes vendidos que figuraban en la tabla maestra).
* **Mantenimiento**. Favoreciendo la simplicidad, definimos que el mantenimiento de un crucero comienza el mismo día en que este es cargado. Es decir, no se pueden realizar mantenimiento programados siendo que el enunciado no lo menciona. Además, definimos como obligatoria la fecha de egreso de mantenimiento, dado que resulta fundamental a nivel negocio para poder programar futuros viajes.
* **Pago de una reserva**. Favoreciendo la simplicidad, definimos que las reservas se deben pagar en forma individual a pesar de que se pueden realizar varias reservas en una única operación, dado que el enunciado no menciona explícitamente que se requiera de otro modo.
* **Rutas de viaje inválidas**. Al momento de realizar la compra de un pasaje, se listan las rutas de viajes disponibles para los filtros aplicados por el usuario. Se valida que no contengan un recorrido inválido y que el crucero esté en vigencia. Decidimos no poblar la tabla de rutas de viaje con un flag de inválido, dado que depende de información presente en otras tablas y sería redundante.

ASPECTOS GENERALES

* En todos los casos, realizamos validaciones para restringir los datos ingresados por el cliente. Por ejemplo, omitiendo espacios a izquierda y derecha, realizando las comparaciones siempre en mayúsculas, entre otras.
* Cuando en el modelado de datos definimos en las descripciones de algunos campos “debe ser único” esto es una decisión asumida por el grupo dado que, si bien el enunciado no lo menciona explícitamente, entendemos favorece a la coherencia de los datos.
* Para las vistas de los ABMs implementamos una partial class en la que definimos una sola estructura para todos los ABMs y mediante constructor de cada formulario en particular utiliza o no ciertas propiedades del formulario que hereda (Form\_Base\_Index).

# PANTALLAS DE LA APLICACION

INGRESO AL SISTEMA



VISTA DEL ADMINISTRADOR



VISTA DEL CLIENTE

