

Pontificia Universidad Católica de Valparaíso
Escuela de Ingeniería Industrial

Documentación de plataforma tecnológica OpenGate y diseño
de propuestas para su implementación.

Informa Final

Por

Renzo Piero Alberti González
Guillermo José Krebs Lopetegui

Prof. Guía: Rosa González R.
Prof. Co-Guía: Guillermo Bustos R.

Mayo, 2014

Índice

Glosario	4
Lista de abreviaturas.....	5
Lista de Figuras	6
Lista de tablas	8
Resumen	9
1 Introducción.....	10
1.1 Objetivos y alcance del Proyecto.....	12
2 Contexto y antecedentes generales.....	13
2.1 Antecedentes comercio exterior Global y en Chile	13
2.1.1 Puerto de San Antonio	18
2.1.2 San Antonio Terminal Internacional	19
2.2 Revisión literaria y Benchmark	20
2.2.1 Revisión de artículos.....	20
2.2.2 Revisión de soluciones activas	22
3 Descripción del Sistema OpenGate	27
3.1 ¿Qué es OpenGate?.....	27
3.2 Objetivos de OpenGate.....	28
3.3 Límite del Sistema	29
4 Documentación de OpenGate en UML	31
4.1 Diagramas UML: Dimensión Estática.....	32
4.1.1 Diagrama de Clases (DCla)	32
4.2 Diagramas UML: Dimensión Funcional	34
4.2.1 Diagrama de Casos de Uso (DCU).....	34
4.2.2 Documentación de Casos de Uso (DoCU)	36
4.3 Diagramas UML: Dimensión Dinámica.....	56
4.3.1 Diagramas de Actividades (DAct).....	56
4.3.2 Diagramas de Máquinas de Estado (DME)	76
4.3.3 Diagramas de Secuencia (DSec).....	82
4.3.4 Diagrama de Interacción Global (DIG)	94
5 Formulación de propuestas para Proyecto II.....	96

5.1 Detalle de propuesta de mayor interés	99
6 Conclusiones.....	102
Bibliografía.....	103
Anexo 1 – Clase Reserva.....	104
Anexo 2 – Clase Usuario Externo	109
Anexo 3 – Clase Transporte	110
Anexo 4 – Clase Gate	111
Anexo 5 – Clase Nave	113
Anexo 6 – Clase Faena	114
Anexo 7 – Clase Controlador	116

Glosario

Faena: Conjunto de actividades logísticas necesarias para que un contenedor entre o salga de la zona primaria (puerto).

Gate: Puerta de entrada a un terminal de carga.

Pallet: Estructura de agrupación de carga fabricado de madera.

Periodo de Stacking: Periodo de tiempo en el cual el terminal marítimo recibe carga contenedora para un determinado servicio naviero (48 horas idealmente).

Stakeholders: Corresponden a los grupos de interés relacionados con la actividad portuaria.

TEU: Unidad de medida que representa la capacidad de carga de un contenedor normalizado de 20 pies.

Visación: Validación aduanera para el ingreso de contenedores al puerto.

Lista de abreviaturas

STI = San Antonio Terminal Internacional.

COLSA = Comunidad Logística de San Antonio.

DCU = Diagrama de Casos de Uso

DCIa = Diagrama de Clases.

DME = Diagrama de máquinas de estados.

DoCU = Documentación de Casos de Uso.

DIG = Diagrama de Interacción Global.

DSec = Diagrama de Secuencia.

DAct = Diagrama de Actividades.

CEPAL = Comisión Económica para América Latina.

EPSA = Empresa Portuaria de San Antonio.

EPA = Empresa Portuaria de Arica.

TPA = Terminal Puerto Arica.

VBS = Vehicle Booking Systems.

TAS = Truck Appointment Systems.

Lista de Figuras

Figura 2.1	– Crecimiento anual del volumen del comercio mundial de bienes y servicios.	13
Figura 2.2	– Composición del transporte marítimo internacional por tipo de carga.	15
Figura 2.3	- Tonelaje transferido por tipo de carga en STI.	20
Figura 2.4	– Distribución de llegadas por ventana horaria, Terminales DP World y Patricken el puerto de Sydney.	24
Figura 2.5	– Ingreso promedio de camiones a STI por ventana horaria (viernes).	25
Figura 2.6	– Ingreso promedio de camiones a STI por ventana horaria (lunes).	25
Figura 3.1	– Logotipo OpenGate.....	28
Figura 3.2	– Menú principal Prototipo OpenGate.	30
Figura 4.1	– Diagrama de Clases Persistente.	33
Figura 4.2	– Diagrama de Clases Dinámico.....	34
Figura 4.3	– Diagrama de Casos de Uso OpenGate.	36
Figura 4.4	– Escenario alternativo OpenGate.	37
Figura 4.5	– Diagrama de Actividades Administrar Capacidad Gate.	58
Figura 4.6	– Diagrama de Actividades Administrar Faenas.....	60
Figura 4.7	– Diagrama de Actividades Crear Faenas.	61
Figura 4.8	– Diagrama de Actividades Habilitar Faenas.....	62
Figura 4.9	– Diagrama de Actividades Habilitar Naves.....	64
Figura 4.10	– Diagrama de Actividades Administrar Reservas.	66
Figura 4.11	– Diagrama de Actividades Ver Faenas Activas.....	68
Figura 4.12	– Diagrama de Actividades Realizar Reserva.....	69
Figura 4.13	– Diagrama de Actividades Crear Usuario Externo.....	71
Figura 4.14	– Diagrama de Actividades Crear Usuario Terminal.....	72
Figura 4.15	– Diagrama de Actividades Ver Planificación.....	73
Figura 4.16	– Diagrama de Actividades Ver Reservas Vigentes.	75
Figura 4.17	– DME Faenas.....	77
Figura 4.18	– DME Gate.	78
Figura 4.19	– DME Nave.	79
Figura 4.20	– DME Reserva.....	80
Figura 4.21	– DME Transporte.	81
Figura 4.22	– DME UsuarioExterno.	81
Figura 4.23	– Clase OpenGateUserDetails.....	82

Figura 4.24 – Diagrama de Secuencia Administrar Capacidad de Gate.	83
Figura 4.25 – Diagrama de Secuencia Administrar Faenas.	84
Figura 4.26 – Diagrama de Secuencia Crear Faena.	85
Figura 4.27 – Diagrama de Secuencia Habilitar Faena.	86
Figura 4.28 – Diagrama de Secuencia Habilitar Nave.	87
Figura 4.29 – Diagrama de Secuencia Administrar Reserva.....	88
Figura 4.30 – Diagrama de Secuencia Ver Faenas Activas.....	89
Figura 4.31 – Diagrama de Secuencia Realizar Reservas.	90
Figura 4.32 – Diagrama de Secuencia Crear Usuario Externo.....	91
Figura 4.33 – Diagrama de Secuencia Crear Usuario Terminal.....	91
Figura 4.34 – Diagrama de Secuencia Ver Planificación.....	92
Figura 4.35 – Diagrama de Secuencia Ver Reservas Vigentes.	94
Figura 4.36 – Diagrama de Secuencia Ver Reservas Vigentes.	95
Figura 5.1 – Situación actual Proyecto OpenPort.	96
Figura 5.2 – Propuestas de Proyecto II en OpenPort.....	97
Figura 5.3 - Beneficios con el uso de sistemas de coordinación y gestión portuaria.	100

Lista de tablas

Tabla 2.1- Transporte marítimo internacional por tipo de carga a nivel mundial (1985-2011) en millones de toneladas métricas anuales	14
Tabla 2.2 – Ranking movimiento portuario de TEUs en América Latina y el Caribe.	16
Tabla 2.3 - Carga de comercio movilizada por puertos Chilenos de propiedad Estatal, en toneladas métricas.....	18
Tabla 2.4 – Cuadro Resumen artículos revisados.....	21
Tabla 2.5 – Cuadro Resumen soluciones revisadas.....	22
Tabla 2.6 – Cuadro de Sistemas VBS y TAS.	23
Tabla 3.1 – Objetivos, indicadores y brechas que abarca el sistema de gestión de transporte.	29
Tabla 4.1 – Herramientas y su utilización para construcción de DCIa.....	32
Tabla 4.2 -Herramientas y su utilización para construcción de DCU.	35
Tabla 4.3 – Actores Externos OpenGate.	35
Tabla 4.4 - Documentación de Caso de Uso “Administrar Capacidad Gate”.	38
Tabla 4.5 - Documentación de Caso de Uso “Administrar Faenas”.....	40
Tabla 4.6 - Documentación de Caso de Uso “Crear Faenas”.....	42
Tabla 4.7 - Documentación de Caso de Uso “Habilitar Faenas”.....	43
Tabla 4.8 - Documentación de Caso de Uso “Habilitar Nave”.	44
Tabla 4.9 - Documentación de Caso de Uso “Administrar Reservas”.	46
Tabla 4.10 - Documentación de Caso de Uso “Ver Faenas Activas”.....	48
Tabla 4.11 - Documentación de Caso de Uso “Realizar Reserva”.....	49
Tabla 4.12 - Documentación de Caso de Uso “Crear Usuario Externo”.....	51
Tabla 4.13 - Documentación de Caso de Uso “Crear Usuario Terminal”.....	53
Tabla 4.14 - Documentación de Caso de Uso “Ver Planificación”.....	54
Tabla 4.15 - Documentación de Caso de Uso “Ver Reservas Vigentes”.	55
Tabla 4.16 - Herramientas y su utilización para construcción de DAct.	56
Tabla 4.17 - Herramientas y su utilización para construcción de DME.....	76
Tabla 4.18 - Herramientas y su utilización para construcción de DSec.	82
Tabla 5.1 – Tabla resumen de propuestas para Proyecto II.	98

Resumen

El presente trabajo desarrollado por Renzo Alberti y Guillermo Krebs para la segunda entrega de Proyecto I, consiste en la documentación descriptiva y conceptual de la plataforma colaborativa de reservas OpenGate, sistema desarrollado para la Empresa Portuaria de San Antonio, específicamente para San Antonio Terminal Internacional.

En la primera parte del trabajo, se presentan los antecedentes mundiales y nacionales, a modo de demostrar la necesidad de plataformas colaborativas VBS en la actualidad, en la segunda parte del trabajo se realiza un benchmark con soluciones activas en el mercado, después se procede a entregar el marco en el que se trabajó para la documentación del prototipo OpenGate, junto con los modelos desarrollados hasta la fecha, finalmente, se presentan las posibles alternativas a seguir para el ramo de Proyecto II.

El modelado del sistema de información, fue desarrollado utilizando el estándar UML, soportado por una herramienta CASE (Visual Paradigm 10.1 Standard Edition). Se ha utilizado de éste lenguaje los modelos, DCU, DAct, DCla, DSec, DME, y DIG, junto con la documentación complementaria DoCU, integrados debidamente para visualizar de mejor manera el comportamiento del sistema OpenGate.

1 Introducción

En la actualidad el comercio exterior marítimo mundial ha aumentado de gran manera, debido a distintos factores como el crecimiento de la economía, las emergentes tecnologías de información, competencia, etc., creciendo a tasas superiores al 10% promedio anual en los últimos 20 años, impulsado por la globalización de los mercados y nuevas tendencias en la gestión logística portuaria. Junto con esto, el volumen de transferencia de carga a nivel nacional no es ajeno a esta tendencia, ya que ha crecido en niveles significativos. Chile ha registrado una fuerte recuperación del comercio exterior transportado por vía marítima, con una participación de un 90% sobre el total de cargas de comercio exterior. Estos impactos positivos se deben a la adaptación a los grandes cambios y tendencias mundiales, que hoy en día, marcan un crecimiento sesgado del comercio en carga contenedorizada.

En esta marcada tendencia al transporte de carga contenedora (TEUs), los terminales portuarios, especializados en la atención de este tipo de carga, presentan un rol estratégico fundamental al constituirse como los nodos que concentran los flujos de carga de importación y exportación de las distintas cadenas logísticas de comercio exterior. Ante un aumento significativo de la demanda de servicios en el puerto y su entorno inmediato, los operadores de terminales han debido generar una serie de iniciativas que permitan una mayor integración con sus clientes y usuarios, donde a través del intercambio de información se logre una reducción en costos y mayor eficiencia operacional.

Sin embargo, estas iniciativas no han sido tan fáciles de implementar, en los últimos años, se ha evidenciado notoriamente la falta de coordinación en el sistema portuario, producto del ya mencionado incremento en los volúmenes de carga exportada y el aumento de las naves, lo que conlleva a un progresivo aumento de viajes desde y hacia el terminal portuario para solventar la demanda y oferta. Las consecuencias son la gran polución y congestión vehicular que afectan a las ciudades y zonas cercanas al puerto, que normalmente sobrepasan las capacidades de apoyo de las instituciones municipales para contener el flujo creado, generando colapsos y saturación inclusive en el mismo terminal.

Cada día existen mayores exigencias para lidiar con el cambio, en este caso, la gestión de operaciones portuarias tienen una enorme cantidad de actores externos, sus stakeholders directos son transportistas terrestres, organismos fiscalizadores, agentes aduaneros, generadores de carga en general, las ciudades en las cuales los puertos interactúan, los trabajadores internos del sistema, etc., todos estos actores tanto públicos como privados están presentes en los cambios que la empresa propone, lo que vislumbra una necesidad latente de plataformas colaborativas de coordinación que permitan satisfacer las necesidades logísticas de estos actores, incrementando la competitividad y eficiencia del sistema portuario en su conjunto.

En respuesta a los crecientes problemas de congestión de camiones dentro y fuera de las puertas de acceso al puerto, muchos operadores de terminales portuarios, en los principales puertos del mundo, están implementando sistemas de control de acceso (vía cámaras o plataformas tecnológicas) o sistemas de reservas (Truck Appointment Systems). Bajo estos sistemas, los usuarios del terminal pueden hacer reservas para sus atenciones durante ventanas

de tiempo específicas para cada faena. El operador de terminal portuario a su vez, debe limitar el número de camiones que pueden ingresar al puerto acorde con la capacidad de atención por faena en cada ventana horaria, tal que los flujos de camiones sean más balanceados durante el día y reducir los periodos *peak* que conllevan a congestión en accesos y dentro del terminal. Las cargas de trabajo predecibles permiten una mejor programación de recursos internos de patio, mayor productividad de las puertas de acceso y mejor uso de los activos de las empresas de servicios de transporte.

Actualmente, en Chile, estos sistema TAS, no se han aplicado a la mayoría de los puertos, debido a que en primera instancia, no se manejan una gran cantidad de carga (en comparación con países como Brasil, Panamá, EEUU), solo el 8% de la carga total de comercio exterior a nivel de América Latina y el Caribe (CEPAL, 2012), por lo que el aumento de carga contenedora ha desarrollado una congestión en ciudades pero no tan alarmante como en otros países, sin embargo, poco a poco se ha notado una mayor necesidad de este tipo de plataformas en los puertos chilenos. Específicamente en el Puerto de San Antonio no cuentan con un sistema de reserva de atenciones terrestres para toda la carga, y solamente las operaciones de despacho de carga de importación directa (referido como despacho directo diferido) se realizan bajo un esquema de citas. Particularmente para el caso de la carga de exportación, los transportistas pueden presentarse al terminal a entregar su carga (cumpliendo previamente con los requisitos de visación) en cualquier momento dentro de la ventana de tiempo habilitada para el stacking de la correspondiente nave y viaje en el que será embarcada la carga. Lo anterior implica que el terminal portuario tenga poco control sobre las llegadas de la carga y por lo tanto el sistema se vuelve ineficiente, con varios períodos del día con recursos ociosos y otros, donde se excede la capacidad de atención. A su vez, este comportamiento genera ineficiencias a los operadores de transporte que pueden experimentar tiempos de espera innecesarios si se presentan durante las horas *peak* principalmente, implicando altos costos de oportunidad.

En el marco de la evaluación para la eventual aplicación de las soluciones anteriormente descritas, el año 2013, con el financiamiento de Innova Corfo, se desarrolló un proyecto denominado OpenPort cuyo objetivo fue el diseño, prueba de conceptos y desarrollo de un prototipo para una solución tecnológica que permita gestionar las reservas de atención a medios de transporte en las terminales portuarias para las operaciones logísticas de recepción y despacho en el terminal portuario, generando a su vez, una herramienta que permita optimizar la cadena de transporte terrestre de contenedores a través de la reducción de fletes en vacío. Lo anterior con la finalidad de contribuir a la competitividad y eficiencia de las cadenas logísticas y de transporte de comercio exterior, mediante la coordinación eficiente de operaciones entre las comunidades de transportistas terrestres, generadores de carga y operadores de terminales portuarias. La solución propuesta desarrollada la Pontificia Universidad Católica de Valparaíso (la beneficiaria y ejecutora) tiene dos principales sistemas componentes que a continuación se describen:

- a) Sistema OpenGate. Sistema para la gestión de reservas para atenciones a medios de transporte en un terminal portuario.
- b) Sistema OpenFleet. Sistema para la comunidad de transportistas del puerto que tiene como objetivo minimizar los fletes en vacío, optimizando de esta manera la cadena de transporte.

En este contexto es donde se sumerge el trabajo de la presente tesis, frente a la solución desarrollada, existe una brecha muy grande entre los modelos lógicos realizados y el prototipo final que se entregó a finales del año 2013, año en el cual el proyecto quedó inactivo por falta de financiamiento, por esto es que se hace necesario el nuevo desarrollo de documentación descriptiva y conceptual de esta plataforma colaborativa, con el fin de volver a trabajar en ella cuando el proyecto vuelva a tener prioridad, además que OpenGate fue hecho de manera modular, es decir su código se formuló para ser flexible frente a nuevas propuestas de mejoras, por lo que se corrobora la necesidad de este trabajo.

La documentación se desarrolló en el Lenguaje Unificado de Modelado, más conocido por su nombre en inglés (*Unified Modeling Language*, UML), es el lenguaje de modelado de sistemas de información y software más conocido y utilizado en la actualidad. Este es un lenguaje gráfico que crea, visualiza, especifica, construye y documenta un sistema. Se estandariza los diagramas de las distintas dimensiones o perspectivas, bajo el paradigma de la orientación a objetos. A continuación se muestran los objetivos y alcances del proyecto.

1.1 Objetivos y alcance del Proyecto

1. Estudiar el contexto general en donde el prototipo OpenGate trabaja, definiendo sus límites.
2. Documentar descriptiva y conceptualmente el prototipo OpenGate en estándar UML con la herramienta CASE visual paradigm.
3. Definir nuevas propuestas de mejoras para la implementación de la plataforma.

Para efectos de estos objetivos, no se considerarán las plataformas OpenFleet y OpenPort. El alcance que tendrá Proyecto I será todo lo correspondiente a la documentación de la plataforma colaborativa, llegando a tener un entregable definitivo en la fecha final de éste, para luego proceder a establecer las directrices para el segundo semestre académico en el ramo de Proyecto II.

2 Contexto y antecedentes generales

En esta sección, se presentan los antecedentes del comercio exterior en un escenario global, continental y nacional, de manera de entender mejor las directrices de las operaciones portuarias, y vislumbrar la problemática de los puertos al abordar una mayor cantidad de exportación e importación de carga contendedora, para luego presentar el Puerto San Antonio como entorno principal del proyecto, en conjunto con la descripción total de la plataforma de reservas OpenGate.

2.1 Antecedentes comercio exterior Global y en Chile

Hoy en día, el comercio marítimo internacional ha experimentado un gran crecimiento, el comercio marítimo internacional experimentó un alza correspondiente a un 5,1% en el año 2011, lo que significa que las 8.678 millones de toneladas transportadas por mar el año 2010, aumentaron a 9.118 millones de toneladas en 2011. A continuación se presenta la evolución del transporte marítimo durante los últimos 26 años (1985-2011), distinguiendo los diversos tipos de productos. Se aprecia que en 2011 el transporte internacional de los cinco principales graneles agrícolas y minerales alcanzó los 2.459 millones de toneladas métricas transportadas, los llamados *minor bulks* alcanzaron los 854, los contenedores los 1.444, mientras que el total de cargas secas ascendió a 5.984 millones de toneladas transportadas; por su parte el petróleo alcanzó un total de 2.857, compuesto por crudo 2.025 y derivados 832, mientras que el gas totalizó 2773.

En la figura 2.1 se puede apreciar cómo ha evolucionado el comercio mundial de bienes y servicios, después del año 2009, se puede ver una tendencia al crecimiento progresivo pero suave en los años posteriores.



Figura 2.1 – Crecimiento anual del volumen del comercio mundial de bienes y servicios. (Fuente: CEPAL, 2012)

El año 2009 el movimiento del comercio exterior en todos sus ámbitos sufrió una contracción respecto al año anterior, 12% en materia de comercio exterior, donde 8,53% corresponde al movimiento portuario de contenedores, las razones de dicha contracción se deben a la crisis económica mundial.

A esto se le puede sumar las crecientes libertades en cuanto a políticas transaccionales en el mundo, lo que da un impulso en los últimos años a fomentar el comercio como una manera de ayudar al desarrollo de los países, esto en materia tangible se traduce en una disminución de los aranceles aduaneros, los que en muchos casos han dejado de tener un impacto sustancial en los costos, además del crecimiento de la industria marítima, cambio que ha sido gatillado por nuevas tecnologías y avances estratégicos, estos avances tecnológicos favorecen de manera sustancial el transporte marítimo de carga contenedora o TEUS, en la Tabla 2.1 se puede apreciar cómo se ha favorecido el transporte de TEUS y su comparación con las otras formas de transportar carga y como efectivamente, durante los últimos 26 años se observa un notable desempeño de los contenedores, cuando su transporte se expandió a una tasa media de 8,8% anual, más que duplicando el promedio del transporte del total de las mercancías. Asimismo, se encuentran ocupando lugares destacados, el gas natural licuado, el carbón térmico, el mineral de hierro y, en resumen, los cinco principales graneles agrícolas y minerales.

Tabla 2.1- Transporte marítimo internacional por tipo de carga a nivel mundial (1985-2011) en millones de toneladas métricas anuales. (Fuente Boletín FAL Edición N° 314, 2012).

Año	MINERALES (hierro, carbón y bauxita)	Graneles agrícolas, mineros y granos	Pequeñas cargas a granel	Contenedores	Otras cargas secas	Total carga seca	Crudo y derivados del petróleo	Gas LNG y LPG	Total (MMtm)
1985	641	1.113	561	160	549	2.170	1.399	61	3.630
1990	739	1.116	606	246	625	2.468	1.603	88	4.159
1995	856	1.318	681	389	727	2.899	1.860	103	4.862
2000	1.017	1.575	749	628	931	3.619	2.174	143	5.936
2001	1.052	1.603	767	647	910	3.667	2.228	143	6.038
2002	1.097	1.665	861	709	954	3.920	2.210	149	6.279
2003	1.178	1.737	906	788	938	4.104	2.352	161	6.617
2004	1.300	1.877	973	917	897	4.391	2.486	169	7.046
2005	1.405	1.982	1.011	1.019	870	4.609	2.576	180	7.365
2006	1.494	2.104	1.084	1.135	831	4.864	2.687	200	7.751
2007	1.620	2.255	1.149	1.263	762	5.127	2.764	210	8.101
2008	1.711	2.388	1.134	1.321	805	5.325	2.760	215	8.300
2009	1.748	2.402	1.035	1.191	743	5.054	2.659	221	7.934
2010	1.974	2.673	1.159	1.342	824	5.660	2.758	260	8.678
2011	2.089	2.793	1.227	1.444	854	5.984	2.857	277	9.118

La nueva tendencia al crecimiento de las exportaciones de TEUs no es ajena a América Latina y el Caribe, para vislumbrar de manera eficiente esta directriz, se presenta la Figura 2.2 que muestra la composición del transporte marítimo internacional por tipo de carga, en donde claramente la carga contenedora ha quitado espacio a otras cargas, principalmente pallets.

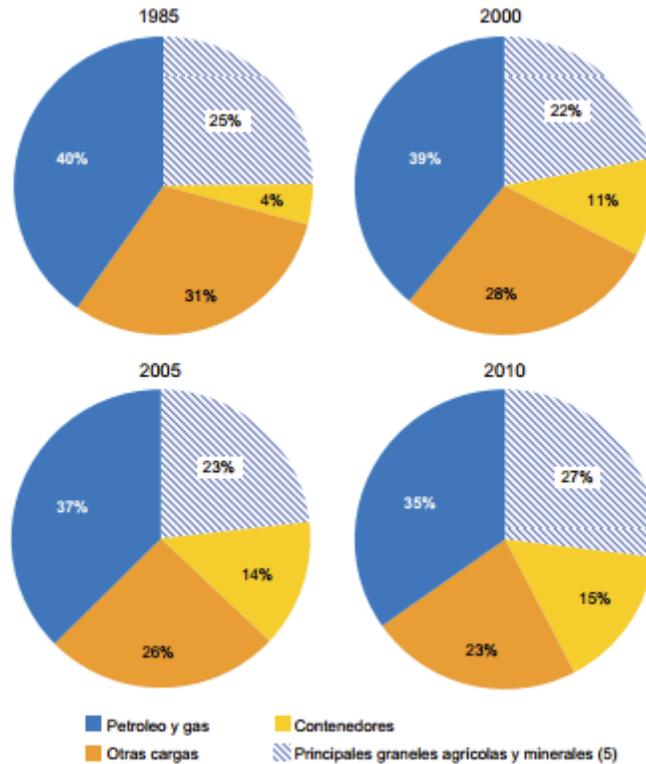


Figura 2.2 – Composición del transporte marítimo internacional por tipo de carga. (Fuente: CEPAL, 2012)

En 2011, los países de América Latina y el Caribe se mantuvieron en el orden del 7% de los contenedores operados en puertos de todo el mundo, alcanzando una cifra record de 41,3 millones de TEU. Sin embargo, la velocidad de crecimiento es diferente entre países y puertos, según ilustra la Tabla 2.2. Del total de la región, el 19.1% es manejado por Brasil, seguido por Panamá (16%), México (10%), Chile (8%), Colombia y Argentina (5%). Si se compara la evolución de la carga durante el primer semestre de 2012 es menos favorable que en años anteriores, desde un 16% de variación interanual durante el primer semestre de 2011 respecto al mismo periodo de 2010, la cifra se redujo a un 8% en el primer medio año del 2012 respecto al de 2011. Durante la primera mitad de 2012, los puertos con el mejor ranking fueron Colón (todas las terminales) y Balboa (Panamá), Santos (Brasil) y la Bahía de Cartagena (Colombia). Cabe destacar los 2 puertos Chilenos en el ranking (Tabla 2.2), con el Puerto de San Antonio en el 8vo lugar y el Puerto de Valparaíso en el puesto número 15, teniendo a San Antonio en un mejor panorama considerando la variación producida entre el 1° semestre del 2012 y el 1°

semestre del 2013 de un 9,9%, siendo así el Puerto con mayor crecimiento entre los primeros 10 puestos.

A continuación se presentará la Tabla 2.2 que muestra el ranking de los 20 puertos de América Latina y el Caribe que movilizaron mayor cantidad de TEUs, en los primeros semestres de los años 2010-2013.

Tabla 2.2 – Ranking movimiento portuario de TEUs en América Latina y el Caribe. (Fuente: CEPAL, 2013)

Rnk 1 Sem.2013	PUERTO	PAIS	1 SEM. 2010	1 SEM. 2011	1 SEM. 2012	1 SEM. 2013	Var. 2013/12
1	Colón	Panamá	1.256.887	1.560.459	1.769.773	1.675.989	-5,30%
2	Santos	Brasil	1.206.532	1.387.624	1.499.703	1.604.478	7,00%
3	Balboa	Panamá	1.209.312	1.549.680	1.671.928	1.592.768	-4,70%
4	Manzanillo	México	669.500	824.417	943.891	1.035.457	9,70%
5	Cartagena	Colombia	699.972	900.411	1.008.938	958.280	-5,00%
6	Callao	Perú	581.121	755.072	870.751	885.411	1,70%
7	Kingston	Jamaica	866.523	792.097	853.214	864.156	1,30%
8	Guayaquil	Ecuador	491.579	683.444	782.933	788.794	0,70%
9	San Antonio	Chile	441.550	484.459	550.520	604.906	9,90%
10	Freeport	Bahamas	600.000	600.000	600.000	600.000	0,00%
11	Buenos Aires	Argentina	515.701	544.784	495.800	549.300	10,80%
12	Caucedo	República Dominicana	450.000	450.000	513.008	545.512	6,30%
13	Puerto Limón-Moin	Costa Rica	469.180	472.548	517.058	535.867	3,60%
14	Lazaro Cárdenas	México	355.519	427.648	586.091	532.726	-9,10%
15	Valparaíso	Chile	442.474	531.661	497.695	485.734	-2,40%
16	Veracruz	México	312.340	356.045	379.818	434.874	14,50%
17	Montevideo	Uruguay	306.441	365.143	376.699	402.440	6,80%
18	Buenaventura	Colombia	294.322	250.293	399.880	387.633	-3,10%
19	Paranaguá	Brasil	334.700	326.544	372.635	358.002	-3,90%
20	Puerto Cabello	Venezuela	197.125	321.478	407.914	350.785	-14,00%

Los mayores transportes de cargas se realizan en el canal de Panamá, lo que se vislumbra en la Tabla 2.2 donde Colón (MIT, evergreen) ocupa el primer puesto, aunque Brasil sea el país con mayor transporte de carga contenedora.

San Antonio ocupó la novena posición en la edición del “Ranking de movimiento portuario contenedorizado” (Tabla 2.2), que anualmente entrega la Comisión Económica para América Latina y el Caribe (CEPAL), respecto al movimiento portuario en la región. De acuerdo

al estudio que se basa en cifras del primer semestre de 2013, el terminal sanantonino movilizó un 9,9% más de TEUs respecto al periodo anterior, lo que lo convierte en el puerto chileno mejor ubicado en dicho listado.

En Chile la costa se caracteriza por ser abierta y de poca profundidad, en donde la necesidad de fomentar el comercio en años pasados ayudó a que se pueda lograr construir puertos en la extensión del país en donde se aprovechan las bahías disponibles. La infraestructura y la logística se han desarrollado acorde a las necesidades generadas de producto movilizado.

En el país, se puede hacer una clasificación de los puertos según criterios:

- i. Según su propiedad pueden ser: puertos estatales y puertos privados. Entendiendo como puertos estatales aquellos cuyas instalaciones pertenecen al Estado de Chile y puertos privados aquellos cuyas instalaciones son de propiedad privada.
- ii. Según su uso pueden ser: puertos de uso público y puertos de uso privado. Entendiendo como puerto de uso público aquel que presta servicios indistintamente a cualquier usuario que lo requiera y que constituye una actividad independiente no accesoria a la industrial principal de su propietario. Por otra parte, se entiende como puerto de uso privado a aquel que ofrece un servicio exclusivo destinado a responder a las necesidades de la actividad principal de su propietario.
- iii. Según su destino o propósito: puertos comerciales, industriales, deportivos, pesqueros, militares, etc. Entendiendo que puerto comercial es todo aquel que tiene por misión fundamental ofrecer a los buques, las instalaciones necesarias para efectuar las operaciones de embarque y desembarque de mercancías y personas.

En Chile existen 10 empresas portuarias del Estado, según lo dispuesto en la Ley 19.542s, que administran los 10 puertos comerciales estatales de uso público del país, y hay 13 empresas portuarias privadas que administran los 14 puertos comerciales de uso público del país.

En la Tabla 2.3 se presentan los puertos estatales y sus respectivas estadísticas por carga de comercio exterior movilizada en los años 2011 y 2012.

Tabla 2.3 - Carga de comercio movilizada por puertos Chilenos de propiedad Estatal, en toneladas métricas. (Fuente: Cámara Marítima y Portuaria de Chile A.G., 2013).

Empresas portuarias del estado*	% movilizado conc.	CARGA GENERAL CONTENED.	CARGA GENERAL FRACC.	CARGA REFRIG. CONTEND.	CARGA REFRIG. FRACC.	CARGA A GRANEL SOLIDOS Y LIQUIDOS	TOTAL CARGA 2012 (tm)	TOTAL CARGA 2011 (tm)
ARICA	100%	1.390.357	68.808	15.905	0	783.559	2.243.430	2.243.430
IQUIQUE	83%	1.286.511	434.594	4.972	209	1.769.677	3.692.869	3.692.869
ANTOFAGASTA	98%	1.103.387	743.756	3.918	47	1.922.109	3.040.411	3.040.411
COQUIMBO	71%	1.120	127.947	23.136	113.244	829.840	688.483	688.483
VALPARAISO	89%	5.487.360	1.132.405	1.477.029	411.050	35.417	8.457.122	8.457.122
SAN ANTONIO	99%	6.093.313	1.152.596	1.715.492	19.085	3.394.188	11.392.263	11.392.263
TALCAHUANO		110	14.442	23	0	92.976	4.443	4.443
SAN VICENTE	99%	3.398.893	171.387	317.195	39.183	2.636.962	5.097.922	5.097.922
PUERTO MONTT		35	10.713	24	0	463.788	472.733	472.733
CHACABUCO		0	0	0	0	0	10.386	10.386
PUNTA ARENAS		1.803	91.052	24	66	191.380	140.072	140.072
SEGÚN CAMARA MARITIMA								
	2012	18.762.835	3.947.700	3.557.718	582.884	12.119.896	38.971.033	
	2011	17.099.868	3.213.149	3.457.175	596.415	10.873.527	35.240.134	35.240.134
TOTAL CARGA	2012	22.344.853	9.828.024	3.818.381	656.112	62.168.532	98.815.902	
COMERCIO EXT.	2011	21.421.906	7.244.526	3.873.096	663.972	61.918.868		95.122.368

2.1.1 Puerto de San Antonio

La Empresa Portuaria San Antonio (EPSA) es una empresa autónoma del Estado, Rol Único Tributario N°61.960.100-9, creada por Ley N° 19.542, sobre modernización del sector portuario estatal, en calidad de continuadora legal de la Empresa Portuaria de Chile (EMPORCHI). Su único propietario es el Estado de Chile y no participa en la propiedad de otras empresas filiales ni coligadas. Está dotada de patrimonio propio, su duración es indefinida y se relaciona con el Gobierno a través del Ministerio de Transportes y Telecomunicaciones.

San Antonio cuenta con cuatro frentes de atraque: Molo Sur, Costanera Espigón, Norte y Policarpo Toro, que suman en total nueve sitios administrados por distintas concesiones:

- El frente de atraque Molo Sur, operado por STI es el que tiene mayor ocupación territorial, el que en sus doce años de concesión ha hecho una gran inversión en infraestructura portuaria, convirtiéndolo en el mejor equipado en la costa este del pacífico sur
- El frente de atraque Costanera Espigón, Administrado por el Terminal Puerto Central, es un Terminal dedicado a todo tipo de carga (multipropósito). El nuevo concesionario tiene el compromiso de construir un frente de atraque continuo de 900 metros, además de la adquisición de equipos para la transferencia y administración de contenedores en patio de última generación.
- El Terminal Norte, Puerto Panul, está dedicado a la transferencia de gráneles sólidos. El terminal ocupa de manera regular la transferencia de carga vía ferroviaria, a través del servicio que presta la empresa de porteo FEPASA.
- El frente de atraque Policarpo Toro, actualmente administrado directamente por EPSA, se especializa en la transferencia de graneles líquidos y cuenta con un sistema de ductos que permite el traslado de carga desde la nave hacia estanques en un sector contiguo.

2.1.2 San Antonio Terminal Internacional

Como fue dicho anteriormente la plataforma de reservas on-line OpenGate está enfocada a resolver la problemática de colas de camiones para el puerto de San Antonio, específicamente en San Antonio Terminal Internacional (STI), es por esto que es necesario presentar los antecedentes de la empresa y su participación en Chile.

STI es hoy en día el principal terminal marítimo del puerto San Antonio. Se desenvuelve dentro del sector marítimo portuario, atendiendo a las empresas navieras y a los exportadores e importadores nacionales, principalmente relacionados a la minería, sector hortofrutícola, forestal, granelero y vitivinícola entre otros. Los principales competidores que enfrenta STI en San Antonio, son Puerto Central S.A., sociedad que inició sus operaciones el 7 de noviembre de 2011 en el sector espigón del Puerto de San Antonio y la Terminal Norte concesionada por Puerto Panul S.A. A nivel regional, también se compite con los otros puertos privados o fiscales que operan en la Región (Valparaíso, Ventanas).

Durante el año 2012 STI transfirió 12.7 millones de toneladas, lo que equivale a un 39,3% de participación en el mercado regional. En términos de volumen de contenedores STI transfirió 639.723 boxes, lo que equivale a un 52.6% de participación en el mercado regional. (Fuente: Memoria anual STI, 2013).

De acuerdo con las Bases de Licitación, STI es una empresa cuyo objeto exclusivo es el desarrollo, mantención y explotación del Frente de Atraque Molo Sur del Puerto de San Antonio, incluyendo la prestación de servicios de muellaje y almacenamiento en el Frente de Atraque, en la Figura 2.3 se presenta el tonelaje transportado transferido por tipo de carga entre

los años 2000 y 2012, cabe mencionar que el cuadro graneles incluye carga sólida y líquida (como gas LNG y petróleo).

Dentro de las principales actividades que desarrolla STI se destacan los servicios de transferencia de carga desde y hacia las naves, servicios de muellaje a la carga y las naves, y otros servicios relacionados, como el almacenamiento de cargas en el terminal y sus actividades complementarias.

Año	Tonelaje Transferido (por tipo de carga)			Total
	Contenedores	Graneles (1)	Carga Fraccionada	
2000	3.076.492	965.553	359.791	4.401.836
2001	2.895.469	1.167.093	389.485	4.452.047
2002	3.328.621	1.687.800	251.645	5.268.066
2003	4.113.148	1.781.287	260.784	6.155.218
2004	5.442.069	1.689.565	150.848	7.282.482
2005	6.349.914	1.673.009	116.972	8.139.895
2006	5.417.216	1.974.827	185.328	7.577.371
2007	5.003.407	2.410.527	479.928	7.893.862
2008	6.210.940	2.413.602	553.769	9.178.311
2009	6.247.421	2.044.283	333.133	8.624.836
2010	7.854.437	2.172.305	636.448	10.663.190
2011	8.815.551	2.111.586	691.410	11.618.547
2012	10.619.183	1.743.635	299.158	12.661.976

Figura 2.3 - Tonelaje transferido por tipo de carga en STI. (Fuente: Memoria anual STI, 2013).

2.2 Revisión literaria y Benchmark

En esta sección se revisaron distintas publicaciones, revistas y artículos de los distintos puertos del mundo que tratan sobre la problemática en sus respectivos Gates, como abordan ésta, y que soluciones se plantean, estas publicaciones serán resumidas en una tabla descriptiva. Para el correcto entendimiento de las problemáticas de eficiencia en los terminales portuarios, se procedió a analizar 6 artículos con distintas visiones, soluciones e implementaciones frente a problemas como la incertidumbre, colapsos en el terminal, ineficiencia en las operaciones de contenedores, integración de sistemas, uso eficiente de espacios, etc., además se analizaron soluciones existentes en el mercado relacionadas a la gestión del Gate, presentando finalmente un Benchmarking.

2.2.1 Revisión de artículos

En la Tabla 2.4 se mostrará un cuadro resumen de los artículos analizados de Gestión de operaciones portuarias, en donde se proporciona de manera sencilla las problemáticas que se abordaron y las implementaciones heurísticas o soluciones junto con sus resultados.

Tabla 2.4 – Cuadro Resumen artículos revisados.

Título	Autor	Año	Resumen
Managing truck arrivals with time windows container terminal.	Chen	2013	Frente al aumento de carga contenedora en terminales portuarios de reducido espacio, se analiza la solución TAM: <i>Truck Arrival Management</i> usando el método “ <i>Vessel Dependent Time Windows (VDTWs)</i> ”.
Inland container depot integration into logistic networks.	Mwemezi	2012	En respuesta a la creciente congestión de los puertos debido a los aumentos de la exportación, se plantea como solución la construcción de un ICD: <i>Inland Container Depot</i> , depósitos físicos afuera del puerto.
Evaluating impact of truck announcements on container stacking efficiency.	van Aspere	2011	Mediante modelos de simulación se intenta cuantificar el impacto en la implementación del sistema TAS: <i>Truck announcement system</i> a los distintos algoritmos y estrategias de apilamiento de contenedores.
Heuristic-based truck scheduling for inland container transportation.	Zhang	2010	Para el problema based <i>truck scheduling</i> se propone una nueva heurística WPB que minimiza el tiempo de alcanzar una solución factible.
Assesing Truck delays using webcams.	Huynh	2011	Mediante un software estadístico se identifican las distribuciones que mejor representan los tiempos entre llegada, tiempo de proceso y tiempo en cola de los camiones en dos terminales.
Performance Analysis of Slot-based Appointment at Air Cargo Terminal	Franz	2012	Mediante simulación se mide el impacto de la implementación de un sistema de reserva para un terminal de carga aéreo.

Esta revisión de artículos muestra las distintas soluciones posibles frente a problemáticas en común, oriundas de la gestión portuaria, y que no son ajenas al Puerto de San Antonio, permiten ver que existe gran flexibilidad en las soluciones, como por ejemplo el método ICD, que propone como se ve en la Tabla 2.4, sobre usar depósitos físicos en áreas externas al puerto,

que pueden ser factibles en ciertos puertos debido a su estructura logística y geográfica, pero quizás en otros no.

Como conclusión de esta revisión de artículos siempre hay que tener en cuenta las variables no controlables del sistema, que acotan distintas soluciones que uno puede formular, el microentorno y macroentorno, el tamaño del puerto, y sus respectivos stakeholders condicionan de manera tangible el marco en donde las propuestas operan.

2.2.2 Revisión de soluciones activas

La idea de la presente sección es el análisis de casos de innovación en cadenas logísticas portuarias, evaluando las posibles soluciones y las mejores prácticas (benchmarking), se presentan resumidos un conjunto de estudios y soluciones revisados por los tesisistas que cubren los ámbitos de negocio foco de este proyecto I+D: Sistemas de gestión de citas/reservas

Tabla 2.5 – Cuadro Resumen soluciones revisadas.

Solución	Año	Resumen
Port of New Bedford TAS.	2010	Plantea un servicio web integral con participación del Estado y Transportistas para resolver la problemática de colas en horas <i>peak</i> del puerto de New Bedford.
Navis Gate Process Automation.	2012	Se proponen soluciones automatizadas de procesos que hoy en día son manuales, el objetivo de estas mejoras es de manera integral junto con el sistema del puerto, reducir costos operacionales, mejorar la seguridad y aumentar la capacidad de Gate en EEUU, India, Hong Kong, Holanda.
1-Stop (Australia)	2010	1-Stop es un centro de información central que ayuda a los transportistas y a la comunidad logística a llevar a cabo un seguimiento de buques, reservas de atención de vehículos en los terminales (VBS), órdenes electrónicas de entrega de importación (EIDOs), aviso anticipado de la carga a recibir (Pre Reival Advice -PRA), facturación y sistemas de pagos, entre otras Características.
e-modal (Estados Unidos)	2000	Esta solución está diseñada para mejorar la eficiencia y disminuir la congestión en los terminales de contenedores. eModal constituye un punto de encuentro en línea para los múltiples terminales de contenedores y sus usuarios.

Dentro de las soluciones activas que se analizaron, destacan 1-Stop y e-modal, estas soluciones están diseñada para mejorar la eficiencia y disminuir la congestión en los terminales de contenedores. E-modal constituye un punto de encuentro en línea para los múltiples terminales de contenedores y sus usuarios, en cuanto a 1-Stop focaliza la prestación de servicios especializados para los participantes de la cadena de suministro en la importación y exportación de mercancías desde y hacia los puertos australianos.

Es necesario destacar los sistemas VBS y TAS (se pueden ver en la Tabla 2.6), en donde ambos apuntan a prácticamente lo mismo, sistemas de reservas entre transportistas y operadores de terminales, la única gran diferencia entre estos, es que uno provee directamente las reservas al transportista (TAS), y el segundo (VBS) permite la posibilidad de que el transportista elija una zona horaria a su comodidad.

Tabla 2.6 – Cuadro de Sistemas VBS y TAS.

VBS (Vehicle Booking Systems)	TAS (Truck Appointment Systems)
<ul style="list-style-type: none"> • Un sistema de reservas de transporte terrestre (VBS) es una plataforma colaborativa de coordinación de los flujos terrestres desde y hacia una terminal portuario para la entrega y recolección de carga. • Proveer de información anticipada con la finalidad de balancear la demanda y oferta 	<ul style="list-style-type: none"> • Permite hacer citas entre transportistas con los operadores de terminales para facilitar el <i>Gate-in</i> y <i>Gate-out</i> de contenedores. • Permite ampliar el concepto de cita/reserva con un terminal a la comunidad logística, logrando unir a los oferentes de transporte con todos los terminales portuarios dentro de la comunidad.

A continuación se presenta un análisis comparativo de llegadas a Gate entre Puerto Sydney (Australia) y Puerto San Antonio realizado por el equipo del proyecto Solución Tecnológica para la Coordinación de Flujos entre Medios de Transporte, Terminales Portuarias y Generadores de Carga Contenedorizada (P11IDL2-10759) (Informe Consolidado San Antonio, 2013). El objetivo será analizar el comportamiento y resultados operacionales del uso y reglamentación de los sistemas VBS administrados por un tercero de base tecnológica (tendencia) y estimar el comportamiento futuro del sistema VBS en comparación (OpenGate).

Respecto al patrón de llegadas:

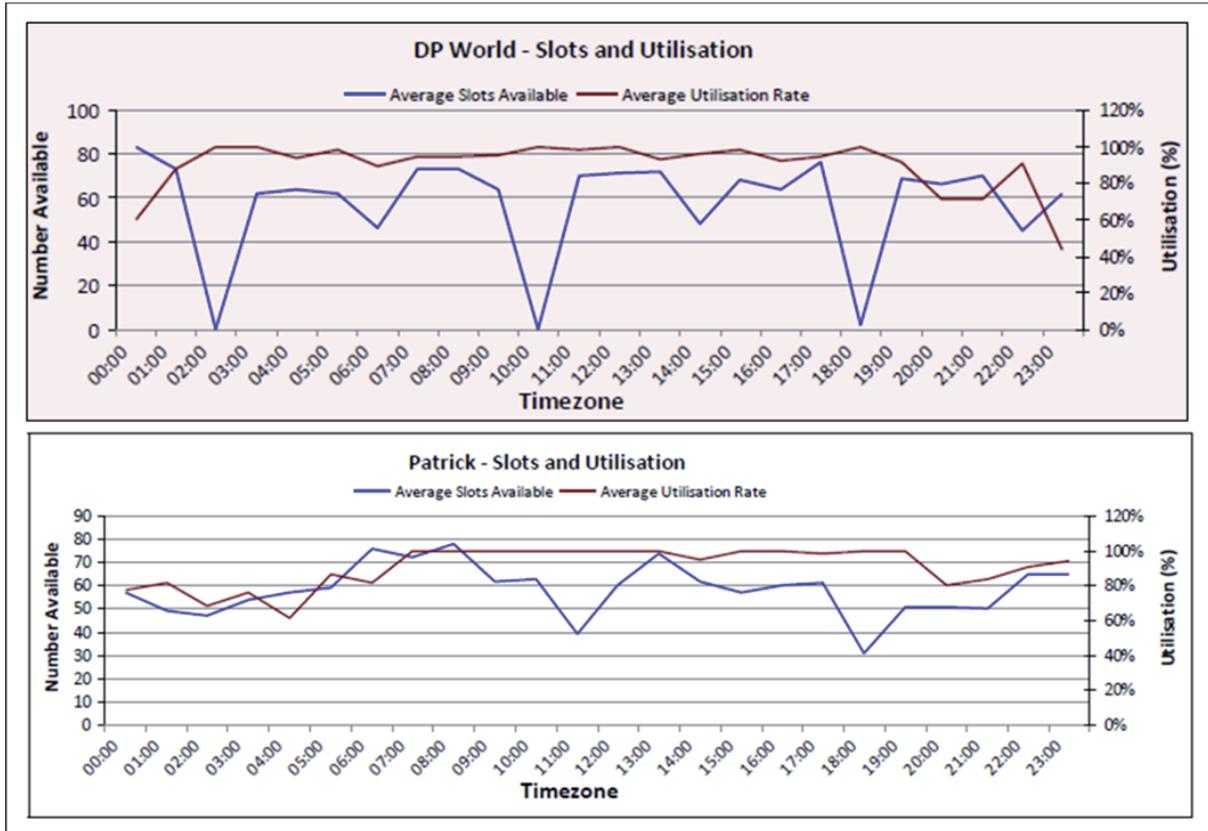


Figura 2.4 – Distribución de llegadas por ventana horaria, Terminales DP World y Patrick en el puerto de Sydney. Fuente: (PBLIS, 2012)

En ambos terminales se presenta una distribución relativamente homogénea en la llegada de camiones, con un mayor nivel de impacto, en los cambios de turno. En el terminal de *DP World*, lo anterior es atribuible a sus políticas y protocolos particulares de atención durante el cambio de turno. El hecho de que la tasa de ingreso se mantenga homogéneamente distribuida durante el turno por zona horaria, refleja el efecto inmediato que generan los sistemas de reserva en la coordinación de llegadas de los transportistas y por consiguiente su ingreso. Ambos terminales tienen implementado el sistema de reserva *I-stop* para la coordinación de flujos de carga, donde todo el universo de usuarios, que deseen ser atendidos en estos terminales, debe registrarse en el sistema y coordinar su llegada a través de una reserva.

Esta modalidad les permite coordinar y gestionar la tasa de ingreso de los camiones a los terminales, en función de la capacidad de atención que presenten los *gates*, procurando no generara situaciones de colapso en las vías de acceso. De las gráficas anteriores se desprende que el nivel de atención máximo de atención, en el terminal de *DP World*, no puede exceder los 70 camiones durante el primer y segundo turno, pudiendo en la noche registrar una mayor tasa de ingreso en una zona horaria determinada. Por otro lado el terminal de Patrick tiene un nivel de ingreso máximo de 80 camiones por hora con un nivel de ocupación del *gate* del 100%.

En el caso de STI, la distribución que presenta el ingreso de camiones está dada por la forma en que se planifican los retiros y entrega de la carga. La Figura 2.5 y 2.6 muestra la distribución por ventana horaria que presenta el ingreso de camiones de entrega y retiro de contenedores en los días viernes y lunes, debido a que los viernes son los días de mayor congestión y los lunes normalmente son días de menor congestión.

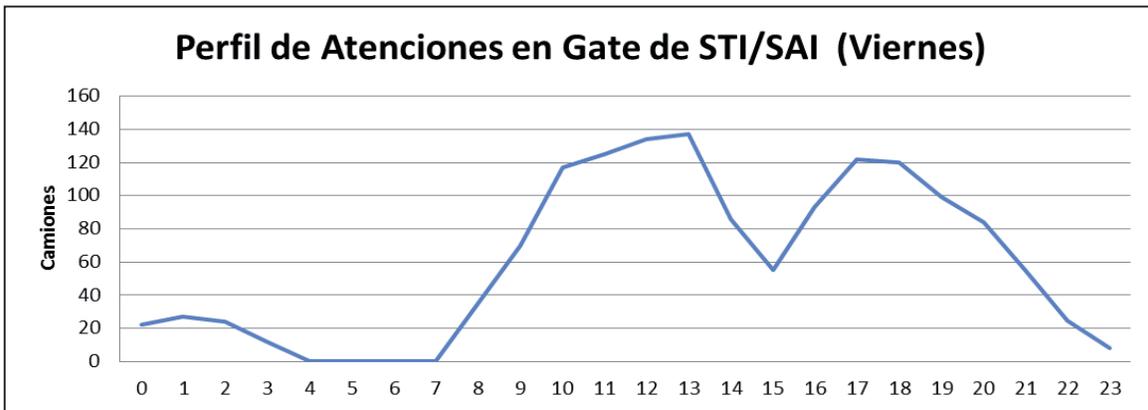


Figura 2.5 – Ingreso promedio de camiones a STI por ventana horaria (viernes). Fuente: (STI, 2012)

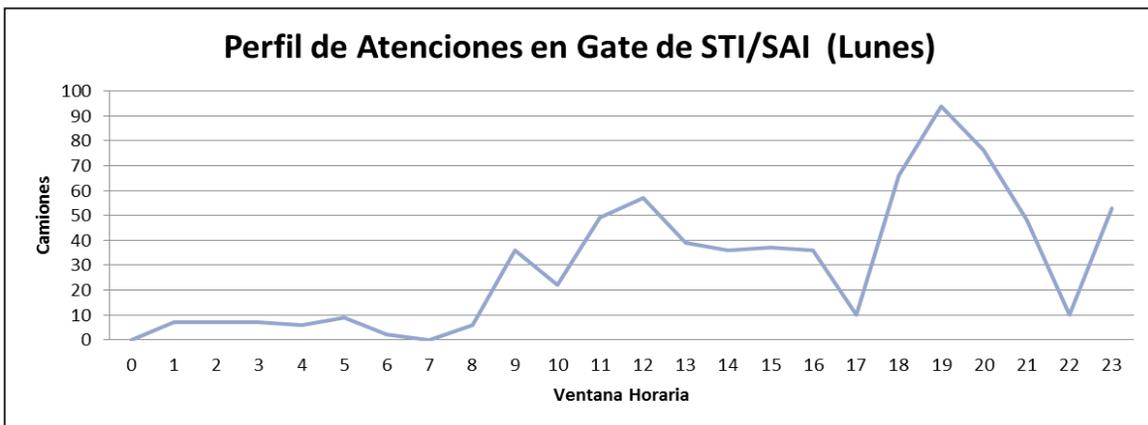


Figura 2.6 – Ingreso promedio de camiones a STI por ventana horaria (lunes). Fuente: (STI, 2012)

Como se puede apreciar existen brechas entre el Terminal y las soluciones estudiadas vistas en las soluciones activas, las principales falencias que presenta el Puerto de San Antonio están asociadas a la falta de coordinación en el ingreso del transporte, al no distribuirse de manera homogénea durante el día, generando horas *peak*, implicando congestión y demoras, que se propagan por el entorno portuario.

Ante esta problemática, se planteó una solución tecnológica que permite mejorar la comunicación entre el Terminal y los transportistas, de manera de lograr una mayor coordinación, con el fin de dar un mejor uso a los tiempos del transportista. La iniciativa en

desarrollo está enfocada en un sistema de citas/reservas para la gestión del volumen de ingreso del transporte en un determinado horizonte temporal, sistema denominado OpenGate.

3 Descripción del Sistema OpenGate

En la sección anterior, se presentó la creciente alza de comercio por tipo de carga contenedora, lo que ha generado una necesidad de coordinación logística y portuaria mayor, en ese marco es donde se mencionó la solución OpenGate. A continuación se explicará en qué consiste la plataforma on-line de reservas OpenGate, sus objetivos, límites y funcionalidades, con el fin de proceder a explicar detalladamente, mediante el uso de estándar UML la plataforma en su completitud.

3.1 ¿Qué es OpenGate?

OpenGate es un sistema del tipo VBS (Vehicle Booking Systems), enfocado en una reserva horaria on-line, el cual pretende mejorar la coordinación para el ingreso de camiones al Terminal, alineando y mejorando la interacción entre los generadores de carga (de forma directa o a través de sus transportistas y/o representantes), responsables de gestionar el envío de camiones para el retiro o entrega de carga, y el operador del terminal responsable de atender a dichos camiones.

Adicionalmente, OpenGate contempla la incorporación de otra tipología de actores, relevantes para el proceso de coordinación de ingreso de transportistas al puerto, que si bien no presentan una interacción directa o “de primera línea”, podrán acceder una serie de funcionalidades asociadas a la visibilidad de la información ingresada por los usuarios principales y realizar operaciones específicas sujetas a ciertas restricciones. Estos usuarios son la autoridad portuaria y los operadores del transporte. OpenGate se enfoca en la alineación de la logística de los generadores de carga y transportistas con la logística de las operaciones del terminal, de tal manera que ambos reciban el beneficio de una menor incertidumbre y de una mejor utilización de sus recursos (menor tiempo de espera del transportista, y mayor capacidad de utilización de espacios y recursos para el terminal).

El sistema tiene como principal foco el mejorar la distribución de llegada de transportistas que llevan carga al Terminal (exportación), dado que como se vio en la sección de brechas, es el flujo que presenta una mayor incertidumbre y variabilidad, complejizando la capacidad de planificación de atención y faenas que realiza el operador portuario.

En la Figura 3.1 se aprecia el logotipo del proyecto:



Figura 3.1 – Logotipo OpenGate. (Fuente: www.corleone.cl/opengate, 2014).

El proyecto se financió por la Línea 2 del Proyecto de I+D de Innova Corfo, siendo la Pontificia Universidad Católica de Valparaíso la beneficiaria y ejecutora del proyecto. A su vez, hay cuatro empresas asociadas al proyecto que corresponden a la autoridad portuaria y el operador de la terminal portuaria en los Puertos de San Antonio y Arica:

- Puerto de San Antonio
 - o Empresa Portuaria de San Antonio, EPSA
 - o San Antonio Terminal Internacional, STI.
- Puerto de Arica
 - o Empresa Portuaria de Arica, EPA
 - o Terminal Puerto Arica, TPA

En primera instancia es importante destacar que no existe actual compromiso con el Puerto de Arica, además, que este proyecto surge como parte de las iniciativas de los respectivos comités de logística de las comunidades de ambos puertos, Comunidad Portuaria de Arica (CPA) y la Comunidad Logística de San Antonio (COLSA), al identificar la falta de coordinación entre el terminal y sus usuarios como la causa principal de los problemas de congestión y demoras para la atención de los mismos.

El proyecto se encuentra en fase de desarrollo, cabe mencionar que existe un avance en el diseño físico del prototipo, disponible on-line en la página mencionada en la Figura 3.1. Debido a dificultades con el financiamiento del proyecto OpenGate en la propuesta del año 2013, es que el presente año 2014 el proyecto se encuentra inactivo.

3.2 Objetivos de OpenGate

Los principales objetivos de OpenGate son reducir tiempo de espera al ingreso del terminal, esto se busca ya que son los transportistas que muchas veces tiene que estar esperando largos periodos de tiempo para ser atendidos en el terminal, lo cual le trae grandes complicaciones, ya que para los transportistas el tiempo es vital, para el negocio, por ejemplo si el transportista está detenido con su camión durante 2 horas, esto significa que esas 2 horas dejó de trabajar, ya que si no tuviese que haber esperado durante tanto tiempo, pudiera ir a buscar otra carga, lo cual le significa otro flete y una mayor ganancia para la empresa; un segundo objetivo que se busca con la implementación de OpenGate es disminuir la incertidumbre sobre

la llegada de camiones al sistema portuario, este objetivo se busca para que el terminal portuario pueda organizarse de mejor manera en los usos de recursos para la atención de estos camiones que reservan su horario de atención; por último se presenta un tercer objetivo, el cual es mejorar la productividad de las faenas portuarias, esto se considera primordial para ambos actores antes mencionados (transportista y terminal), ya que estos podrá aprovechar de mejor manera sus recursos y podrá lograr realizar tareas eficientes y eficaces.

En la Tabla 3.1 se muestran los objetivos específicos que pretende abordar el sistema OpenGate, de manera de mitigar las brechas descritas, y que se verán reflejados en los indicadores de gestión.

Tabla 3.1 – Objetivos, indicadores y brechas que abarca el sistema de gestión de transporte. (Fuente: Informe consolidado San Antonio, 2013).

Objetivo	Brecha	KPI
Aumentar la productividad del camión.	Viajes muertos	Tonelaje transportado/ Kilómetro recorrido.
Identificar en tiempo real oportunidades de negocio.	Viajes muertos	Servicios realizados / Período de Tiempo.
Disminuir tiempo de estadía del transportista en el terminal o su entorno.	Tiempos de rotación de los camiones.	Tiempo de rotación (Tiempo de espera + tiempo de atención) de una importación y exportación.
Disminuir cantidad de camiones en el entorno portuario.	Tiempos de rotación de los camiones y viajes muertos.	Camiones / Día

3.3 Límite del Sistema

Como el prototipo y el código del sistema ya están creados, es por lo cual el modelo tiene un propósito descriptivo, es decir, representa la forma de operar de un sistema ya existente, con éste se pretende asegurar una comprensión acabada del sistema OpenGate. De igual manera correspondería realizar un modelo físico del sistema, pero se optó por representar lo que hace el sistema en un ámbito más cercano a lo lógico que a lo físico.

➤ Límite del sistema

El límite del sistema serán todas las actividades, estructuras, y actores que comprenden el alcance total del prototipo informático OpenGate, disponible vía web en <http://www.corleone.cl/opengate> (Figura 3.2). Por lo que no se considerarán OpenPort, OpenFleet y los servidores de bases de datos de OpenGate. De igual manera debido a que el prototipo es el sistema de software, es que se dejará fuera del sistema al Sistema de negocio que abarca OpenPort.

Cabe mencionar que en el código java del prototipo OpenGate, existen varias estructuras y asociaciones del código que están enfocadas en desplegar por pantalla el prototipo, mantener actualizado los datos y recibir datos vía internet e integrarlos en su base de datos, estos, siguiendo la proposición inicial de lo que se va a documentar, queda fuera del límite que este proyecto abarca.



Figura 3.2– Menú principal Prototipo OpenGate. Fuente: (<http://www.corleone.cl/opengate>, 2014)

En la Figura 3.3 se puede apreciar las interacciones que existen en el sistema, como se puede apreciar existe un modelo que se encarga de interactuar con el usuario y de actualizar la base de datos, por otro lado está el modelo de OpenGate que es el modelo encargado del funcionamiento en sí del prototipo, este modelo ocupa los datos que el modelo de interacción con el usuario integra en las bases de datos.

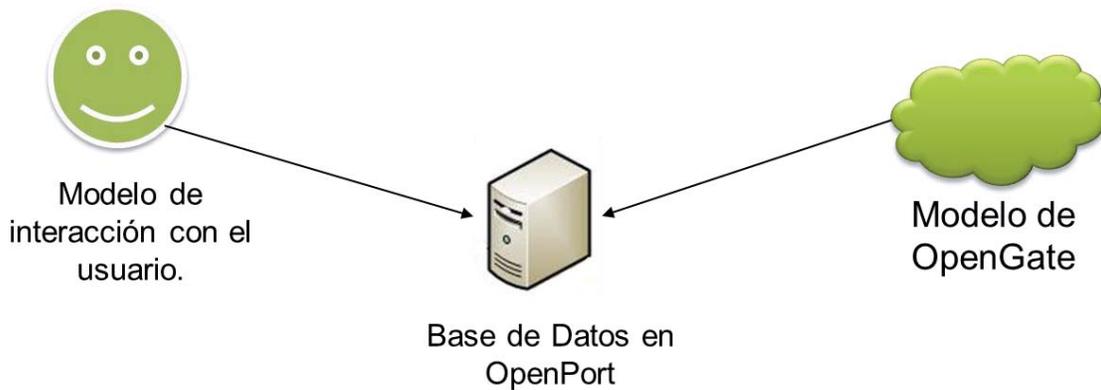


Figura 3.3– Interacciones en el Sistema.

4 Documentación de OpenGate en UML

En esta sección se ven los distintos modelos de las dimensiones Estático, Dinámico y Funcional correspondientes a la documentación en estándar UML, los modelos mostrados a continuación fueron realizados con la herramienta CASE Visual Paradigm. Ya establecidos los límites del sistema y las directrices que tomarán las documentaciones por el equipo, se adjuntan de manera complementaria los supuestos y herramientas que el equipo dispuso para la realización de esta tarea.

Para la realización de esta documentación, el equipo de tesis contó con la plataforma activa vía on-line en la página ya mencionada en la sección 3, junto con esto el manual del usuario respectivo y el código total de la plataforma, que son el suprasistema OpenPort y los subsistemas OpenGate y OpenFleet, todo esto en formato java.

Antes de presentar los modelos finales para el ramo de Proyecto I, es necesario establecer bajo qué criterios y supuestos se modeló el sistema, por lo que a continuación se presentan éstos de manera tal de entender completamente el modelado de la plataforma de reservas. Cabe mencionar que todos los supuestos tomados tienen en consideración un principal objetivo, mostrar de manera clara y completa la esencia del sistema, de manera de facilitar el rápido entendimiento de éste y respetando las tareas fundamentales que cumple.

Decisiones y supuestos adoptados para la documentación:

1. El usuario administrador tiene acceso a todas las funcionalidades pero su rol es de apoyo, por lo que se consideró solo los CU en los cuales tiene participación necesaria para el desarrollo del CU.
2. Se usó un nivel de abstracción mayor en el DClA debido a la enorme cantidad de componentes de este (modelo real, representativo 100% del código java), usando paquetes que en éste proyecto serán considerados Clases.
3. Se dejarán de lado todas las clases de interfaz con el usuario, excepto en el diagrama de clases dinámico y diagrama de secuencia, que se representarán estas en 1 controlador, que es la abstracción de los controladores necesarios para poder explicar detalladamente los procedimientos establecidos en los 12 Casos de Uso.
4. Como OpenPort es el sistema que se actualiza por los servidores de base de datos y OpenGate accede a los datos de OpenPort es que se omitirán todos los procedimientos de actualización e importación de datos del sistema, dejándoselas al actor secundario OpenPort.
5. Debido a que la excepción en los CU de interrupción en la conexión a internet, es igual para todos estos, se omitió éste en los diagramas de actividades de manera de evitar una complejidad innecesaria para el entendimiento del comportamiento del sistema.
6. Se dejarán de lado las clases del código java que no son propias del funcionamiento de OpenGate.
7. Se dejarán de lado las clases del código java que no tienen relaciones ni asociaciones (clases inactivas).

8. Para el Diagrama de secuencias se omitirán los ingresos de usuario y contraseña de los terminales y externos, de manera de no detallar demasiada información sobre registros y enfocarse en la esencia del sistema, cabe mencionar que estas verificaciones corresponden a la clase OpenGateUserDetails que se explica en la sección 4.3.3.
9. Para ayudar a la comprensión inicialmente de la secuencia y comportamiento en el diagrama de secuencia, es que se omitirán en las operaciones sus argumentos, esto en la sección 4.3.3., a modo de complementación se pueden revisar detalladamente en la sección Anexos.
10. Los mensajes en el DSec serán encapsulados de manera que los diagramas sean más simples de entender, ya que el objetivo del diagrama de secuenciación es mostrar la comunicación que existe entre las clases, el proceso de encapsulamiento se llevó a cabo mediante mensajes provenientes de la misma clase emisora con destino a la misma clase receptora.

4.1 Diagramas UML: Dimensión Estática

La dimensión Estática trata justamente del aspecto estático de los sistemas, se enfatiza en representar la estructura del sistema, como la documentación se realizó en estándar UML, corresponde mostrar el Diagrama de Clases.

4.1.1 Diagrama de Clases (DCla)

El DCla se caracteriza por ser un diagrama que muestra la composición estructural de un sistema, muestra que objetos pertenecen al sistema y son permanentes en él, usado para describir clases en aplicaciones de software. Para la realización del correspondiente Diagrama de Clases para OpenGate, el equipo de tesis obtuvo el código java de OpenGate, que representa estructuralmente el prototipo, gracias a la ayuda del equipo de informáticos en I+D de la Universidad, se logró obtener un DCla final y descriptivo del sistema OpenGate, debido a que en el alcance del proyecto se decidió documentar sólo lo que hace el sistema, es que se dejó de lado el código java correspondiente al diseño visual del prototipo y las transformaciones de datos vía internet.

Para la construcción del Diagrama de Clases del proyecto, se presenta la Tabla 4.1, que especifica las herramientas utilizadas para la construcción de los distintos componentes del modelo.

Tabla 4.1 – Herramientas y su utilización para construcción de DCla.

Herramienta	Utilización
Código Java	Clases, métodos y atributos. (Ingeniería Inversa)
Prototipo OpenGate (www.corleone.cl/opengate)	Empaquetamiento en Clases Visibles

Gracias a la herramienta Visual Paradigm 10.2 se pudo trabajar con el código java en código bidireccional, usando la funcionalidad de ingeniería inversa de éste, logrando obtener el diagrama de clases que representa el 100% del código de la plataforma OpenGate, obteniéndose así todo lo relacionado a OpenPort en Completitud, el paso siguiente fue discriminar que corresponde dentro de los límites establecidos y que no, finalmente se llegó a la decisión de mostrar 2 Diagramas de Clases, el primero correspondiente al entendimiento de la estructura tangible del sistema, y el segundo, el necesario para realizar todas las funcionalidades que el prototipo provee, es decir, incluyendo una clase controladora proveniente de la interfaz de interacción con el usuario, cabe mencionar que se empaquetó las clases controladoras en una sola, quedando como clase final la Clase Controlador.

En segunda instancia, para efectos de lograr una mayor comprensión de lo esencial del sistema, es que se presentarán a continuación los modelos de clases sin atributos ni operaciones, con el fin de comprender las asociaciones y relaciones que estos tienen en un nivel de abstracción mayor, por sobre de las invocaciones de operaciones para la funcionalidad interna de las estructuras, en la sección **Anexo** (al final de éste informe), se podrán revisar todas las clases propuestas en estos 2 diagramas, incluyendo las clases internas y sus interacciones, de esta manera no perder el objetivo principal propuesto en Proyecto, que es la documentación descriptiva. A continuación se presentan los dos diagramas de Clases propuestos.

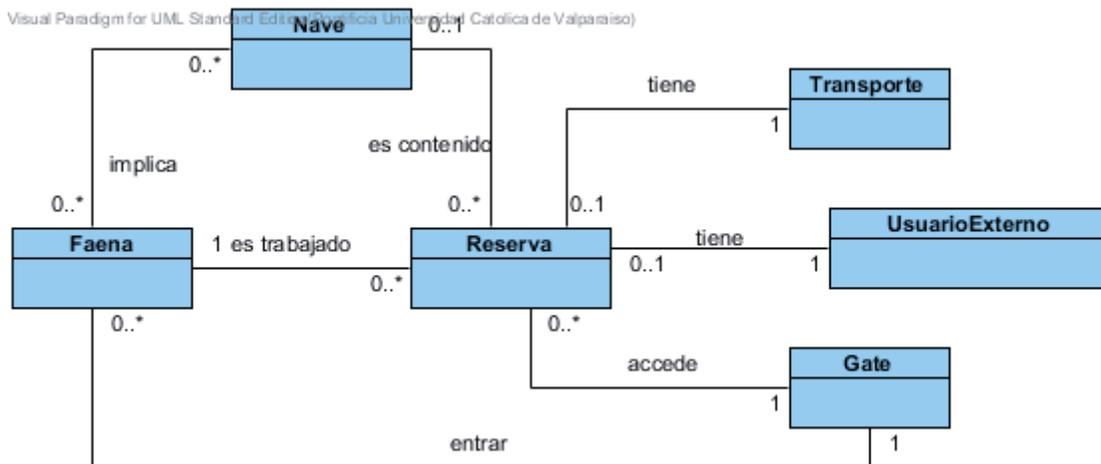


Figura 4.1 – Diagrama de Clases Persistente.

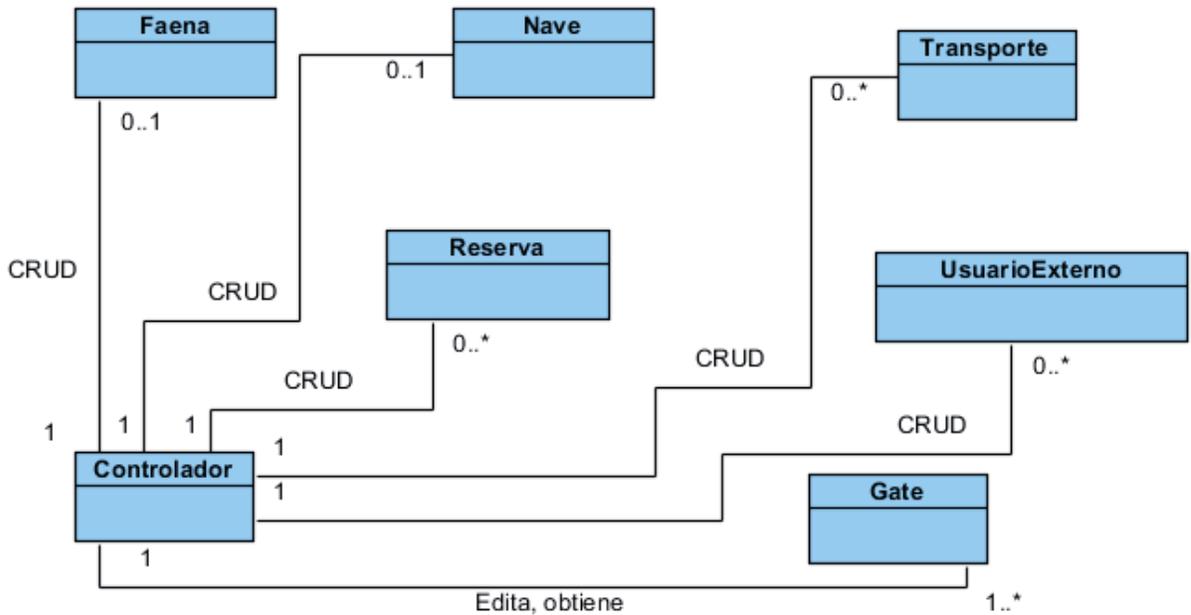


Figura 4.2 – Diagrama de Clases Dinámico.

4.2 Diagramas UML: Dimensión Funcional

La dimensión Funcional trata de la función de conversión que realiza el sistema, se enfatiza en la transformación de entradas en salidas que lleva a cabo el sistema, se indica el qué hace el sistema, como la documentación se realizó en estándar UML, corresponden mostrar el Diagrama de Casos de Uso, junto con su Documentación de Casos de Uso, cabe mencionar que la DoCU no pertenece al aspecto funcional del sistema, sólo permite entender el conjunto de actividades necesarias que realiza cada caso de uso.

4.2.1 Diagrama de Casos de Uso (DCU)

El DCU se caracteriza por ser un diagrama qué muestra que hace el sistema y no el cómo lo hace, estos listan las funcionalidades del sistema pero no indican los procedimientos o actividades necesarios para llevarlos a cabo. Estos son las unidades funcionales del sistema, que generan valor para un actor.

Para la realización de este Diagrama el equipo de tesistas trabajó de manera paralela con tres herramientas importantes de este sistema (Tabla 4.2), estos son: el manual de usuario OpenGate, el prototipo OpenGate y el código Java, los primeros 2 ambos disponibles vía web en: www.corleone.cl/opengate. Con la información obtenida y mediante el procedimiento descendente (Modelos de Sistemas de Información, 2010), se obtuvo el DCU en completitud.

Para la construcción del Diagrama de Clases del proyecto, se presenta la Tabla 4.2, que especifica las herramientas utilizadas para la construcción de los distintos componentes del modelo.

Tabla 4.2 -Herramientas y su utilización para construcción de DCU.

Herramienta	Utilización
Manual de Usuario OpenGate	Escenarios Principales
Prototipo OpenGate (www.corleone.cl/opengate)	Escenarios Alternativos y Excepciones
Código Java	Actores Secundarios y límites del sistema

A continuación se encuentra explicado detalladamente (Tabla 4.3) los actores identificados del sistema OpenGate.

Tabla 4.3 – Actores Externos OpenGate.

ACTORES	DEFINICIÓN
Usuario Externo	Es el usuario reservador, físicamente el transportista que desea reservar una zona horaria para retirar o recibir una carga, es el que recibe la mayoría del valor en las funcionalidades del sistema, debido a que la plataforma está diseñada para facilitar las reservas en ella.
Usuario Terminal	Es el usuario interno del sistema, físicamente es el subordinado a cargo del apoyo del sistema dentro de la empresa, puede reservar en nombre de un usuario externo y definir las zonas horarias por Gate.
Usuario Administrativo	Es el usuario administrativo de apoyo, físicamente es el encargado del funcionamiento de la plataforma, no debería reservar ni realizar funcionalidades de la plataforma pero tiene acceso a la mayoría de estas en caso de que se necesite.
OpenPort	Es el suprasistema de OpenGate, OpenPort tiene clases y métodos que el sistema importa, por lo que se hace necesario este para que OpenGate pueda trabajar.

Finalmente se presenta el Diagrama de Casos de Uso de OpenGate, donde se pueden apreciar 12 Casos de Usos que representan en completitud la funcionalidad del sistema, respecto al Usuario Administrador, se estableció que solo se consideraría en las creaciones de usuario donde tiene un rol fundamental en aceptación del externo y en creación del interno (terminal), como se ve en la tabla anterior (Tabla 4.3), el Usuario Administrador tiene acceso a todas las funcionalidades pero no percibe valor de ninguna, debido a que tiene un rol de apoyo en el

sistema, es por esto que se omitió la participación de éste en los demás Casos de Uso debido a representar de mejor manera la funcionalidad esencial del sistema.

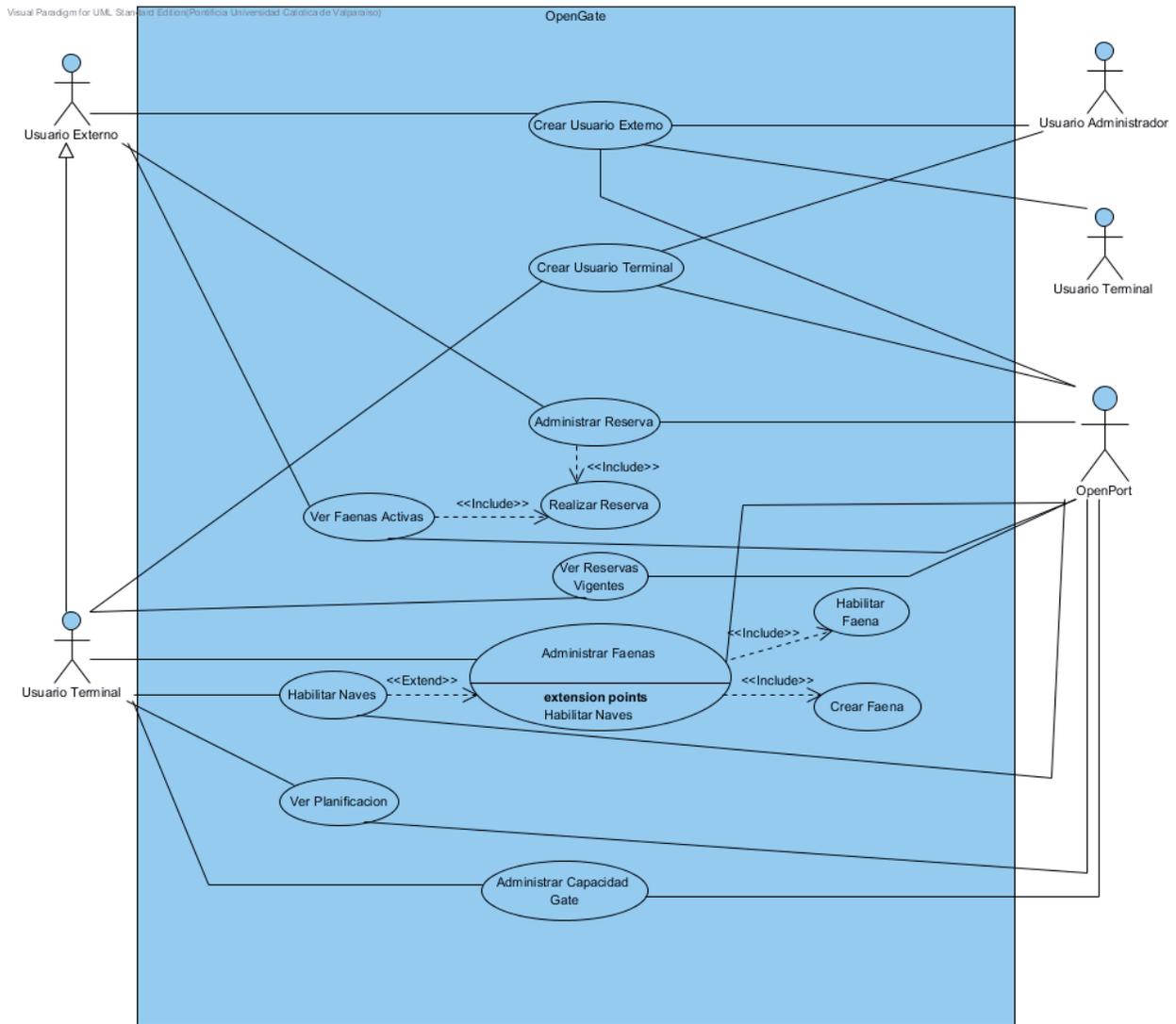


Figura 4.3 – Diagrama de Casos de Uso OpenGate.

4.2.2 Documentación de Casos de Uso (DoCU)

La DoCU como su nombre lo indica, es la documentación detallada de las actividades necesarias que un Caso de Uso realiza para su generación de valor, es decir, muestra que sucede en cada uno de los CU, su rol es complementario y solo tiene sentido en el contexto de un DCU dado. Es importante destacar que la DoCU no forma parte del estándar UML.

Cabe mencionar que para conocer las excepciones y escenarios alternativos se probaron en el prototipo on-line como respondía el sistema frente a distintos escenarios y estímulos, uno

de estos se puede apreciar en la Figura 4.3, en donde se ve cómo reacciona el sistema frente a un error en los campos de contraseña, como también frente a el escenario donde el usuario no rellena los campos.

Registrar Planificador Gate

 **The button/link/text component needs to have a Form in its ancestry. Please add <h:form>.**

Información Planificador Gate

Nombre:

Departamento: Campo requerido

Información de Acceso

Nombre Usuario: Campo requerido

Contraseña: Contraseñas no coinciden. Intente nuevamente.

Confirme Contraseña:

Zona Horaria: ▼

✕ Cancelar
✔ Confirmar

© OpenGate, 2013 - Términos de Uso

Figura 4.4 – Escenario alternativo OpenGate. (Fuente: www.corleone.cl/opengate, 2013).

Finalmente se presentan a continuación las documentaciones de los 12 Casos de Uso contemplados en la sección 4.2.1. En el campo observaciones se pueden ver distintas apreciaciones de los tesisistas con respecto al sistema.

Tabla 4.4 - Documentación de Caso de Uso “Administrar Capacidad Gate”.

Caso de Uso	Administrar Capacidad Gate
Objetivo	Permite al Usuario terminal determinar la capacidad máxima que tendrá la ventana horaria en la semana determinada
Ámbito	Negocio
Actor Principal	Usuario Terminal
Actores Secundarios	OpenPort
Pre-condición	Usuario Terminal Creado
Garantías mínimas	No hay
Post-condición	Se guarda capacidad definida
Inicio	Usuario Terminal
Escenario principal	<p>1 El caso de uso se inicia cuando Usuario terminal desea administrar capacidad de Gate.</p> <p>2 Usuario ingresa nombre de usuario y contraseña.</p> <p>3 Usuario terminal selecciona opción Gate.</p> <p>4 Usuario terminal selecciona opción Capacidad de Gate.</p> <p>5 Sistema accede a OpenPort por información de Gate, Faenas y Naves.</p> <p>6 Sistema despliega por pantalla 7 semanas con información de Gate, Faenas y Naves asociadas.</p> <p>7 Usuario Terminal selecciona Gate y semana de interés.</p> <p>8 Usuario Terminal selecciona opción Nuevo.</p> <p>9 Sistema despliega por pantalla campos de zonas horarias a definir.</p> <p>10. Usuario Terminal selecciona zonas horarias.</p> <p>11. Usuario Terminal selecciona opción Crear.</p> <p>12 Sistema Actualiza OpenPort.</p> <p>13. Sistema informa “se ha creado capacidad de Gate vacía con x zonas horarias correctamente”.</p> <p>14 Sistema Despliega zonas horarias de Gate creado.</p> <p>15. Usuario Terminal selecciona editar.</p> <p>16. Usuario Terminal rellena zonas horarias con capacidad.</p> <p>17 Usuario Terminal selecciona guardar capacidad.</p> <p>18 Sistema Actualiza OpenPort.</p> <p>19 Sistema informa se ha guardado capacidad de Gate.</p> <p>20 CU termina.</p>
Escenarios alternativos	<p>2a Usuario y contraseña no existentes</p> <p>2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2.</p> <p>7a Usuario Terminal desea editar Gate creado.</p> <p>7a1 Usuario Terminal selecciona Gate Creado. Regresa paso 14.</p> <p>15a Usuario Terminal no desea editar capacidad</p> <p>15a1 Regresa paso 20.</p> <p>17a Usuario terminal desea guardar configuración de Gate.</p> <p>17a1 Usuario Terminal selecciona guardar configuración.</p>

	<p>17a2 Sistema despliega campo de nombre de configuración. 17a3 Usuario Terminal ingresa nombre de configuración 17a4 Usuario Terminal selecciona guardar configuración. 17a5 Sistema actualiza OpenPort. 17a6 Sistema informa configuración guardada exitosamente. Regresa paso 17.</p>
Excepciones	<p>*a Se pierde conexión a internet. *a1 Sistema espera retorno de conexión. Regresa paso 20.</p>
Puntos de extensión	No hay
Observaciones	<ol style="list-style-type: none"> 1. Por default los Gate vienen instanciados, pero no definidos. 2. Sistema muestra 4 semanas anteriores a la actual y 2 semanas posteriores a la actual. 3. La selección de Gate en el paso 7 puede ser seleccionando las semanas desplegadas o por los campos a rellenar. 4. La configuración guardada tiene por objetivo que se vuelva a usar en otros Gates, la funcionalidad de cargar la configuración no está habilitada en el sistema.

Tabla 4.5 - Documentación de Caso de Uso “Administrar Faenas”.

Caso de Uso	Administrar Faenas
Objetivo	Permite al Usuario Terminal administrar las faenas para disposición del Usuario Externo
Ámbito	Negocios
Actor Principal	Usuario Terminal
Actores Secundarios	OpenPort
Pre-condición	Usuario Terminal Creado
Garantías mínimas	No hay
Post-condición	Cambios actualizados en OpenPort
Inicio	Usuario Terminal
Escenario principal	<p>1 El caso de uso se inicia cuando Usuario Terminal desea administrar las faenas</p> <p>2 Usuario terminal ingresa nombre de usuario y contraseña</p> <p>HACER CASO</p> <p>CASO Usuario terminal desea registrar faena para coordinar a través de plataforma.</p> <p>3 Incluye Crear Faena</p> <p>CASO Usuario terminal desea habilitar faena existente</p> <p>4 Incluye Habilitar Faena.</p> <p>CASO Usuario Terminal desea editar capacidad de faena</p> <p>5 Usuario Terminal selecciona faena</p> <p>6 Usuario Terminal selecciona habilitar faena.</p> <p>7 OpenPort accede a información de faenas, naves.</p> <p>8 Sistema despliega por pantalla faenas registradas y gates.</p> <p>9 Usuario Terminal selecciona Gate y semana.</p> <p>10 Usuario Terminal selecciona Editar.</p> <p>11 Sistema despliega por pantalla campos de faena y modalidad,</p> <p>12 Usuario Terminal selecciona faena</p> <p>13 Usuario Terminal selecciona opción seleccionar.</p> <p>14 Sistema despliega por pantalla calendario con horarios y capacidades de faena a rellenar.</p> <p>15 Usuario Terminal rellena capacidades en campos</p> <p>16 Usuario Terminal selecciona guardar faena.</p> <p>17 Sistema Despliega mensaje de éxito.</p> <p>FIN CASO</p> <p>18 Actualizar datos en OpenPort</p> <p>19 CU Termina</p>
Escenarios alternativos	<p>2a Usuario y contraseña no existentes.</p> <p>2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2.</p> <p>4a Se creó una faena con requerimiento de especificar nave</p> <p>4a1 Extiende Habilitar Nave. Regresa paso 7.</p> <p>10a Usuario Terminal desea ver faena.</p>

	<p>10a1 Usuario Terminal selecciona Ver.</p> <p>10a2 Sistema despliega por pantalla campos de faena y modalidad.</p> <p>10a3 Usuario Terminal selecciona faena.</p> <p>10a4 Usuario Terminal selecciona opción seleccionar.</p> <p>10a5 Sistema despliega por pantalla calendario con horarios y capacidades de faena. Regresa paso 19.</p>
Excepciones	<p>*a Se pierde conexión a internet.</p> <p>*a1 Sistema espera retorno de conexión. Regresa paso 19</p>
Puntos de extensión	Habilitar Nave
Observaciones	No hay

Tabla 4.6 - Documentación de Caso de Uso “Crear Faenas”.

Caso de Uso	Crear Faena
Objetivo	Permite al Usuario Terminal crear una nueva Faena
Ámbito	Negocios
Actor Principal	Usuario Terminal
Actores Secundarios	OpenPort
Pre-condición	Usuario ingresado, Usuario Terminal Creado
Garantías mínimas	No hay
Post-condición	Nueva Faena Creada
Inicio	Invocado por CU Administrar Faenas
Escenario principal	<p>1 El caso de uso se inicia cuando Usuario terminal desea crear nueva faena.</p> <p>2 Usuario terminal selecciona configuraciones.</p> <p>3 Usuario terminal selecciona tipos de faenas.</p> <p>3 Sistema accede a OpenPort por detalles de faenas</p> <p>4 Sistema muestra el listado de faenas total registradas.</p> <p>5 Usuario terminal selecciona Registrar.</p> <p>6 Sistema dispone los campos necesarios para registrar nueva faena.</p> <p>7 Usuario terminal rellena los campos: Modalidad, Especificación de nave, nombre de faena y elige los atributos para la reserva.</p> <p>8 Usuario terminal selecciona confirmar.</p> <p>8 Sistema crea nueva faena y le incorpora los atributos para la reserva.</p> <p>9 Sistema incorpora nueva faena a “tipos de faenas”.</p> <p>10 Sistema muestra el listado de faenas total registradas.</p> <p>11 CU termina.</p>
Escenarios alternativos	<p>5a Usuario Terminal sólo desea ver detalles de faenas registradas</p> <p>5a1 Usuario Terminal selecciona ver detalle de faena de interés.</p> <p>5a2 Sistema despliega por pantalla detalle de faena (modalidad, nombre y atributos). Regresa paso 11.</p> <p>5b Usuario Terminal desea ver detalles de faenas registradas.</p> <p>5b1 Usuario Terminal selecciona ver detalle de faena de interés.</p> <p>5b2 Sistema despliega por pantalla detalle de faena (modalidad, nombre y atributos).</p> <p>5b3 Usuario Terminal cierra ventana desplegada. Regresa paso 5</p>
Excepciones	<p>*a Se pierde conexión a internet.</p> <p>*a1 Sistema espera retorno de conexión. Regresa paso 11</p>
Puntos de extensión	No hay
Observaciones	No hay

Tabla 4.7 - Documentación de Caso de Uso “Habilitar Faenas”.

Caso de Uso	Habilitar Faena
Objetivo	Crear instancias de las faenas registradas
Ámbito	Negocios
Actor Principal	Usuario Terminal
Actores Secundarios	OpenPort
Pre-condición	Faena Creada, Gate habilitado
Garantías mínimas	No hay
Post-condición	Faena Habilitada
Inicio	Invocado por CU Administrar Faenas
Escenario principal	<ol style="list-style-type: none"> 1 El CU se inicia cuando Usuario Terminal desea habilitar faena. 2 Usuario Terminal selecciona Faenas. 3 Usuario Terminal selecciona Habilitar Faenas. 4 Sistema accede a OpenPort por información de faenas. 5 Sistema muestra Faenas existentes (habilitadas y no habilitadas) y opciones de Gate y semanas. 6 Usuario Terminal selecciona Gate. 7 Usuario Terminal selecciona semana. 8 Usuario Terminal selecciona nuevo. 9 Sistema despliega por pantalla mensaje de habilitación de faena. 10 Usuario Terminal selecciona confirmar. 11 Sistema habilita faena. 12 Sistema muestra mensaje de operación exitosa. 13 CU termina.
Escenarios alternativos	<ol style="list-style-type: none"> 6a No existe capacidad de Gate definida para semana seleccionada. 6a1 Sistema muestra por pantalla mensaje “no ha planificado capacidad de gate, es necesario planificar la capacidad del gate antes de habilitar faenas”. Regresa paso 13.
Excepciones	No hay
Puntos de extensión	No hay
Observaciones	No hay

Tabla 4.8 - Documentación de Caso de Uso “Habilitar Nave”.

Caso de Uso	Habilitar Nave
Objetivo	Habilitar una nave a una faena
Ámbito	Negocios
Actor Principal	Usuario Terminal
Actores Secundarios	OpenPort
Pre-condición	Faena Habilitada
Garantías mínimas	No hay
Post-condición	Nave habilitada
Inicio	Usuario Terminal
Escenario principal	<p>1 El CU se inicia cuando Usuario terminal desea adjudicar Nave a Faena</p> <p>2 Usuario terminal ingresa Usuario y contraseña</p> <p>3 Usuario terminal selecciona opción Faenas</p> <p>4 Usuario terminal selecciona Habilitar Nave</p> <p>5 OpenPort accede a datos de naves.</p> <p>6 Sistema despliega por pantalla todas las naves planificadas por el terminal con información relativa a estas y su fecha de arribo próximo.</p> <p>7 Usuario terminal selecciona asignar faena a nave de interés</p> <p>8 Sistema despliega por pantalla campo de semana a rellenar.</p> <p>9 Usuario Terminal selecciona semana.</p> <p>10 Usuario Terminal selecciona asignar faena.</p> <p>11 Sistema despliega por pantalla información de la nave y campos de faena a rellenar. (Nombre faena, fecha inicio y fecha término.).</p> <p>12 Usuario Terminal rellena campos de faena.</p> <p>13 Usuario Terminal selecciona asignar faena.</p> <p>14 Usuario Terminal selecciona guardar</p> <p>15 Sistema actualiza datos.</p> <p>16 Sistema informa operación exitosa “se ha modificado correctamente para la nave X”.</p> <p>17 Sistema despliega por pantalla todas las naves planificadas por el terminal con información relativa a estas y su fecha de arribo próximo.</p> <p>18 CU termina</p>
Escenarios alternativos	<p>2a Usuario ya identificado en el sistema</p> <p>2a1 Regresa paso 3.</p> <p>2b Usuario y contraseña no existentes</p> <p>2b1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2.</p> <p>7a Usuario Terminal desea ver detalle de la nave.</p> <p>7a1 Usuario Terminal selecciona ver detalle de nave de interés.</p> <p>7a2 Sistema despliega por pantalla campo de semana a rellenar.</p> <p>7a3 Usuario Terminal selecciona ver detalle de faena asociadas.</p>

	<p>7a4 Sistema despliega por pantalla información de nave con tabla de faenas asociadas. Regresa paso 18.</p> <p>13a Usuario Terminal deja campo vacío.</p> <p>13a1 Sistema informa “campo requerido” en campo que no se rellenó. Regresa paso 11.</p> <p>14 Usuario Terminal desea asignar más faenas a la nave.</p> <p>14a1 Regresa paso 11.</p>
Excepciones	<p>*a Se pierde conexión a internet.</p> <p>*a1 Sistema espera retorno de conexión. Regresa paso 18.</p>
Puntos de extensión	No hay
Observaciones	No hay

Tabla 4.9 - Documentación de Caso de Uso “Administrar Reservas”.

Caso de Uso	Administrar Reservas
Objetivo	Permite al Usuario Externo gestionar de manera completa el sistema de reservas.
Ámbito	Negocios
Actor Principal	Usuario Externo
Actores Secundarios	OpenPort
Pre-condición	Usuario Externo Creado, Faena Habilitada
Garantías mínimas	No hay
Post-condición	Reservas administradas
Inicio	Usuario Externo
Escenario principal	<p>1 El caso de uso se inicia cuando Usuario Externo desea gestionar reservas.</p> <p>2 Usuario Externo ingresa usuario y contraseña</p> <p>3 Usuario externo selecciona opción Reservas</p> <p>HACER CASO</p> <p>CASO Usuario Externo desea Realizar Reserva</p> <p>4 Usuario Externo selecciona opción Reservar</p> <p>5 Incluye Realizar Reserva</p> <p>CASO Usuario externo desea ver sus Reservas</p> <p>6 Usuario externo selecciona ver mis Reservas</p> <p>7 Sistema pide semana y Gate</p> <p>8 Usuario ingresa semana y Gate.</p> <p>9 Sistema entrega semana y horario con reservas del Usuario externo.</p> <p>10 Usuario selecciona opción “confirmar”.</p> <p>FIN CASO</p> <p>11 CU termina</p>
Escenarios alternativos	<p>2a Usuario y contraseña no existentes.</p> <p>2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”.</p> <p>9a No existen Reservas en la semana y Gate seleccionados.</p> <p>9a1 Sistema Despliega mensaje de error “no hay reservas para la semana seleccionada”. Regresa paso 8.</p> <p>10a Usuario externo desea modificar datos transportista</p> <p>10a1 Usuario externo selecciona reserva de interés</p> <p>10a2 Usuario selecciona modificar datos transportista.</p> <p>10a3 Sistema despliega campos a rellenar.</p> <p>10a4 Usuario modifica datos de interés</p> <p>10a5 Sistema actualiza Base de Datos. Regresa paso 10.</p> <p>10b Usuario externo desea modificar horario reserva</p> <p>10b1 Usuario externo selecciona nuevo horario manipulando reserva vigente.</p> <p>10b2 Sistema actualiza Base de Datos. Regresa paso 10.</p>

	<p>10c Usuario externo desea cancelar reserva. 10c1 Usuario externo selecciona reserva de interés 10c2 Usuario externo selecciona cancelar reserva. 10c3 Sistema cambia estado de reserva. 10c4 Sistema actualiza Base de Datos. Regresa paso 10.</p>
Excepciones	<p>*a Se pierde conexión a internet. *a1 Sistema espera retorno de conexión. Regresa paso 15</p>
Puntos de extensión	No hay
Observaciones	<p>Modificar horarios de reservar en fechas próximas a la actual puede incluir multas (no está contemplado en OpenGate).</p> <p>La manipulación de las reservas incluye extenderla a más de una zona horaria continua.</p>

Tabla 4.10 - Documentación de Caso de Uso “Ver Faenas Activas”.

Caso de Uso	Ver Faenas Activas
Objetivo	Ver el nivel de ocupación semanal que presentará el Gate por zona horaria
Ámbito	Negocios
Actor Principal	Usuario Externo
Actores Secundarios	OpenPort
Pre-condición	Usuario Externo Creado
Garantías mínimas	No hay
Post-condición	Se despliega calendario
Inicio	Usuario Externo
Escenario principal	<ol style="list-style-type: none"> 1 El caso de uso se inicia cuando Usuario Terminal desea ver las faenas activas 2 Usuario Terminal ingresa usuario y contraseña 3 Usuario Terminal selecciona opción faenas 4 Usuario Terminal selecciona Faenas activas 5 Sistema accede a OpenPort por información de Faenas 6 Sistema despliega calendario con primera Faena de la lista. 7 Sistema pide datos de interés: Gate, Modalidad, Faena y Semana. 8 Usuario Terminal ingresa datos de interés 9 Sistema busca en OpenPort faena que cumplan con datos ingresados 10 Sistema despliega calendario con faena seleccionada. 11 Incluye Realizar Reserva. 12 CU termina.
Escenarios alternativos	<ol style="list-style-type: none"> 2a Usuario y contraseña no existentes. <ol style="list-style-type: none"> 2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2. 6a No existen faena habilitada. <ol style="list-style-type: none"> 6a1 Regresa paso 12. 7a Faena desplegada es faena de interés del Usuario. <ol style="list-style-type: none"> 7a1 Regresa paso 11. 10a No existen faenas con datos ingresados. <ol style="list-style-type: none"> 10a1 Sistema informa “no existen faenas de modalidad seleccionada para la semana”. Regresa paso 7. 11a Usuario Externo no desea realizar reserva <ol style="list-style-type: none"> 11a1 Regresa paso 12.
Excepciones	<p>*a Se pierde conexión a internet.</p> <p>*a1 Sistema espera retorno de conexión. Regresa paso 12</p>
Puntos de extensión	No hay
Observaciones	No hay

Tabla 4.11 - Documentación de Caso de Uso “Realizar Reserva”.

Caso de Uso	Realizar Reserva
Objetivo	Permite al Usuario Externo realizar una reserva en el sistema
Ámbito	Negocios
Actor Principal	Usuario Externo
Actores Secundarios	OpenPort
Pre-condición	Usuario Externo validado
Garantías mínimas	No hay
Post-condición	Reserva Realizada
Inicio	Invocado de CU Administrar Reservas CU Ver Faenas activas
Escenario principal	<p>1 El CU se inicia cuando Usuario Externo desea realizar una reserva</p> <p>2 Sistema accede a OpenPort y obtiene información de Gate y Semana.</p> <p>3 Sistema muestra información.</p> <p>4 Usuario Externo selecciona Semana, Gate y Modalidad.</p> <p>5 Sistema accede a OpenPort por información de Faena.</p> <p>6 Sistema despliega información de Faenas.</p> <p>7 Usuario Externo selecciona Faena.</p> <p>8 Sistema pide información de carga.</p> <p>9 Usuario Externo ingresa información de carga.</p> <p>10 Sistema muestra calendario de la semana con sus horarios.</p> <p>11 Usuario externo elige horario y selecciona confirmar reserva.</p> <p>12 Sistema Actualiza OpenPort.</p> <p>13 Sistema entrega Detalle Reserva = código reserva + reservador + faena + contenedor + fecha reserva + estado reserva.</p> <p>14 CU termina</p>
Escenarios alternativos	<p>6a No existe faena habilitada</p> <p>6a1 Regresa a paso 14.</p> <p>9a Usuario desea incluir datos de transportista</p> <p>9a1 Usuario externo selecciona opción incluir información de transportista.</p> <p>9a2 Sistema despliega campos requeridos a ingresar datos.</p> <p>9a3 Usuario externo ingresa Transportista Camión y Conductor.</p> <p>9a4 Sistema ingresa transportista. Regresa Paso 10.</p> <p>11a Usuario selecciona horario gris, rojo, café o Amarillo.</p> <p>11a1 Sistema no responde. Regresa Paso 11.</p>
Excepciones	<p>*a Se pierde conexión a internet.</p> <p>*a1 Sistema espera retorno de conexión. Regresa paso 7</p>
Puntos de extensión	No hay
Observaciones	1 La modalidad se refiere a si es carga de importación o exportación.

	<p>2 En calendario de la semana con horarios se muestran 5 colores:</p> <p>Gris: Cuando la fecha ya ha pasado.</p> <p>Rojo: Cuando la nave no está habilitada para reservas.</p> <p>Café: Si no hay faenas disponibles para la reserva deseada.</p> <p>Amarillo: Cuando el horario es muy próximo al actual y la reserva solo puede ser llevada a cabo por teléfono.</p> <p>Azul: Muestra la reserva recién efectuada, hecha a través de un click en la zona horaria elegida.</p>
--	---

Tabla 4.12 - Documentación de Caso de Uso “Crear Usuario Externo”.

Caso de Uso	Crear Usuario Externo
Objetivo	Permite al Usuario Terminal Crear usuario externo
Ámbito	Negocios
Actor Principal	Usuario Externo
Actores Secundarios	Usuario Terminal, OpenPort
Pre-condición	Usuario Terminal creado
Garantías mínimas	No Hay
Post-condición	Usuario Externo Creado
Inicio	Usuario Terminal
Escenario principal	<p>1 El CU se inicia cuando Usuario Terminal desea crear un planificador o Usuario Externo.</p> <p>2 Usuario Terminal ingresa Usuario y contraseña.</p> <p>3 Usuario Terminal selecciona Configuraciones</p> <p>4 Usuario Terminal selecciona Solicitar Registro Usuario</p> <p>5 Sistema solicita “información Usuario externo”, “información acceso” y “información ubicación comercial”.</p> <p>6 Usuario Terminal ingresa datos.</p> <p>7 Usuario Terminal selecciona zona horaria de Usuario Externo.</p> <p>8 Usuario Terminal selecciona confirmar.</p> <p>9 Sistema informa de ingreso de solicitud de registro</p> <p>10 Usuario Terminal accede a solicitudes pendientes</p> <p>11 Usuario Terminal Aprueba solicitud de creación de Usuario.</p> <p>12 Sistema crea PerfildeUsuario en OpenPort.</p> <p>13 Sistema actualiza OpenPort</p> <p>14 Sistema despliega por pantalla solicitud aprobada con éxito.</p> <p>15 CU termina.</p>
Escenarios alternativos	<p>2a Usuario y contraseña no existentes.</p> <p>2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2.</p> <p>6a Usuario terminal deja casilla en blanco.</p> <p>6a1 Sistema solicita nuevamente la información, al lado de la casilla en blanco sistema informa “campo requerido”. Regresa paso 6.</p> <p>6b Usuario terminal ingresa usuario existente</p> <p>6b1 Sistema solicita nuevamente la información, al lado de la casilla “nombre Usuario” en blanco sistema informa “campo requerido”. Regresa paso 6.</p> <p>6c Usuario ingresa contraseñas distintas.</p> <p>6c1 Sistema solicita nuevamente la información, al lado de la casilla “contraseña” en blanco sistema informa “contraseñas no coinciden, intente nuevamente”. Regresa paso 6.</p> <p>11a Usuario Terminal rechaza solicitud.</p> <p>13a1 Sistema informa rechazo de solicitud. Regresa paso 15.</p>

Excepciones	*a Se pierde conexión a internet. *a1 Sistema espera retorno de conexión. Regresa paso 15
Puntos de extensión	No hay
Observaciones	Sistema debería mandar notificación a email de usuario externo registrado cuando se valida su cuenta. (No lo hace, solo informa a terminal). Si el usuario terminal creo el usuario reservador, no debería tener que aprobarlo nuevamente

Tabla 4.13 - Documentación de Caso de Uso “Crear Usuario Terminal”.

Caso de Uso	Crear Usuario Terminal
Objetivo	Permite al Usuario Administrativo crear un Usuario Terminal
Ámbito	Negocios
Actor Principal	Usuario Terminal
Actores Secundarios	Usuario Administrador, OpenPort
Pre-condición	Usuario Administrador existente o Usuario Terminal existente
Garantías mínimas	No hay
Post-condición	Usuario Terminal Creado
Inicio	Usuario Administrador o Usuario Terminal
Escenario principal	<p>1 El CU se inicia cuando Usuario Administrador desea crear un planificador o Usuario Terminal.</p> <p>2 Usuario Administrativo ingresa al sistema.</p> <p>3 Usuario Administrativo selecciona Configuraciones</p> <p>4 Usuario Administrativo selecciona Registrar Planificador</p> <p>5 Sistema despliega campos a rellenar obligatorios</p> <p>6 Usuario Administrativo ingresa datos: Nombre + departamento + nombre usuario + contraseña + confirmación contraseña</p> <p>7 Usuario Administrativo selecciona zona horaria de Usuario Terminal</p> <p>8 Usuario Administrativo selecciona confirmar.</p> <p>9. Sistema crea PerfildeUsuario en OpenPort.</p> <p>10 Sistema actualiza Base de Datos</p> <p>12 CU termina.</p>
Escenarios alternativos	<p>2a Usuario y contraseña no existentes.</p> <p>2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2.</p> <p>6a Usuario Administrativo deja casilla en blanco.</p> <p>6a1 Sistema solicita nuevamente la información, al lado de la casilla en blanco sistema informa “campo requerido”. Regresa paso 6.</p> <p>6b Usuario Administrativo ingresa usuario existente</p> <p>6b1 Sistema solicita nuevamente la información, al lado de la casilla “nombre Usuario” en blanco sistema informa “usuario existente, ya existe usuario”. Regresa paso 6.</p> <p>6c Usuario Administrativo ingresa contraseñas distintas.</p> <p>6c1 Sistema solicita nuevamente la información, al lado de la casilla “contraseña” en blanco sistema informa “contraseñas no coinciden, intente nuevamente”. Regresa paso 6.</p>
Excepciones	No hay
Puntos de extensión	<p>*a Se pierde conexión a internet.</p> <p>*a1 Sistema espera retorno de conexión. Regresa paso 12</p>
Observaciones	No se informa de creación de usuario exitosa.

Tabla 4.14 - Documentación de Caso de Uso “Ver Planificación”.

Caso de Uso	Ver Planificación
Objetivo	Permite al Usuario visualizar el estado del sistema
Ámbito	Negocios
Actor Principal	Usuario Terminal
Actores Secundarios	OpenPort
Pre-condición	Usuario Terminal Creado
Garantías mínimas	No hay
Post-condición	Sistema despliega planificación
Inicio	Usuario Terminal
Escenario principal	<p>1 El caso de uso se inicia cuando Usuario Terminal desea ver la situación actual del estado del sistema</p> <p>2 Usuario Terminal ingresa usuario y contraseña</p> <p>3 Usuario Terminal selecciona opción faenas</p> <p>4 Usuario Terminal selecciona planificación</p> <p>5 Sistema despliega campos a rellenar de Gate y semana</p> <p>6 Usuario Terminal ingresa Gate y semana</p> <p>7 Usuario Terminal seleccionar “seleccionar”.</p> <p>8 Sistema accede a OpenPort por información de Faenas, capacidad de faenas y capacidad de Gate,</p> <p>9 Sistema despliega por pantalla una planilla donde se indica planificación general y el nivel de ocupación por zona horaria, incluyendo un campo de faena posible a rellenar.</p> <p>10 CU Termina</p>
Escenarios alternativos	<p>2a Usuario y contraseña no existentes</p> <p>2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2.</p> <p>9a Usuario ingreso Gate que no ha sido definido su capacidad para semana ingresada.</p> <p>9a1 Sistema muestra un mensaje indicando que no existe planificación. Regresa paso 6.</p> <p>10a Usuario Terminal desea ver planificación por faena específica</p> <p>10a1 Usuario Terminal selecciona faena en particular</p> <p>10a2 Sistema despliega por pantalla planilla con planificación particular y nivel de ocupación por zona horaria. Regresa paso 10.</p>
Excepciones	<p>*a Se pierde conexión a internet.</p> <p>*a1 Sistema espera retorno de conexión. Regresa paso 10</p>
Puntos de extensión	No hay
Observaciones	Por default en el paso 9 se entrega nivel de ocupación de todas las faenas habilitadas.

Tabla 4.15 - Documentación de Caso de Uso “Ver Reservas Vigentes”.

Caso de Uso	Ver Reservas Vigentes
Objetivo	Permite al Usuario Terminal ver genéricamente todas las reservas vigentes
Ámbito	Negocios
Actor Principal	Usuario Terminal
Actores Secundarios	OpenPort
Pre-condición	Usuario Terminal Creado
Garantías mínimas	No hay
Post-condición	Se despliega información de reservas en día seleccionado
Inicio	Usuario Terminal
Escenario principal	<p>1 El CU se inicia cuando Usuario terminal desea ver las reservas vigentes.</p> <p>2 Usuario Terminal ingresa usuario y contraseña</p> <p>3 Usuario Terminal selecciona Reservas</p> <p>4 Usuario Terminal selecciona ver reservas vigentes</p> <p>5 Sistema despliega por pantalla campo de semana a rellenar</p> <p>6 Usuario Terminal selecciona semana de interés</p> <p>7 Usuario Terminal selecciona “seleccionar”</p> <p>8 Sistema accede a OpenPort por información de Reservas y Faenas.</p> <p>9 Sistema despliega por pantalla información de faenas habilitadas por día de la semana junto con las reservas realizadas, canceladas y ejecutadas.</p> <p>10 Usuario Terminal selecciona ver detalle de día de interés.</p> <p>11 Sistema despliega por pantalla todas las reservas de día seleccionado y campos de faena y estado reserva a rellenar.</p> <p>12 CU termina.</p>
Escenarios alternativos	<p>2a Usuario y contraseña no existentes</p> <p>2a1 Sistema informa “Error de autenticación, intenta de nuevo. Motivo: Bad credentials”. Regresa paso 2.</p> <p>9a Usuario Terminal selecciona semana no existen reservas vigentes.</p> <p>9a1 Sistema despliega por pantalla mensaje “no hay reservas para la semana seleccionada”. Regresa paso 6.</p> <p>10a Usuario Terminal no desea ver detalle de día específico.</p> <p>10a1 Regresa paso 12.</p> <p>12a Usuario desea ver reservas de faena y estado específico.</p> <p>12a1 Usuario Terminal rellena campo de faena y estado.</p> <p>12a2 Usuario Terminal selecciona filtrar reservas.</p> <p>12a3 Sistema despliega por pantalla reservas del día en estado y faena seleccionado.</p> <p>12a4 Usuario Terminal selecciona volver. Regresa paso 12.</p> <p>12b Usuario Terminal desea ver detalle de reserva.</p> <p>12b1 Usuario Terminal selecciona ver detalle reserva.</p> <p>12b2 Sistema despliega detalle reserva.</p>

	12b3 Usuario Terminal selecciona volver. Regresa paso 12.
Excepciones	*a Se pierde conexión a internet. *a1 Sistema espera retorno de conexión. Regresa paso 12.
Puntos de extensión	No hay
Observaciones	En el paso 5 se rellena por default la semana actual

4.3 Diagramas UML: Dimensión Dinámica

La dimensión Dinámica trata del comportamiento del sistema, se enfatiza en la representación de cómo se comporta el sistema en el tiempo, como la documentación se realizó en estándar UML, corresponden mostrar los Diagramas de Actividades, Diagramas de Máquinas de Estados y el Diagrama de Interacción Global.

4.3.1 Diagramas de Actividades (DAct)

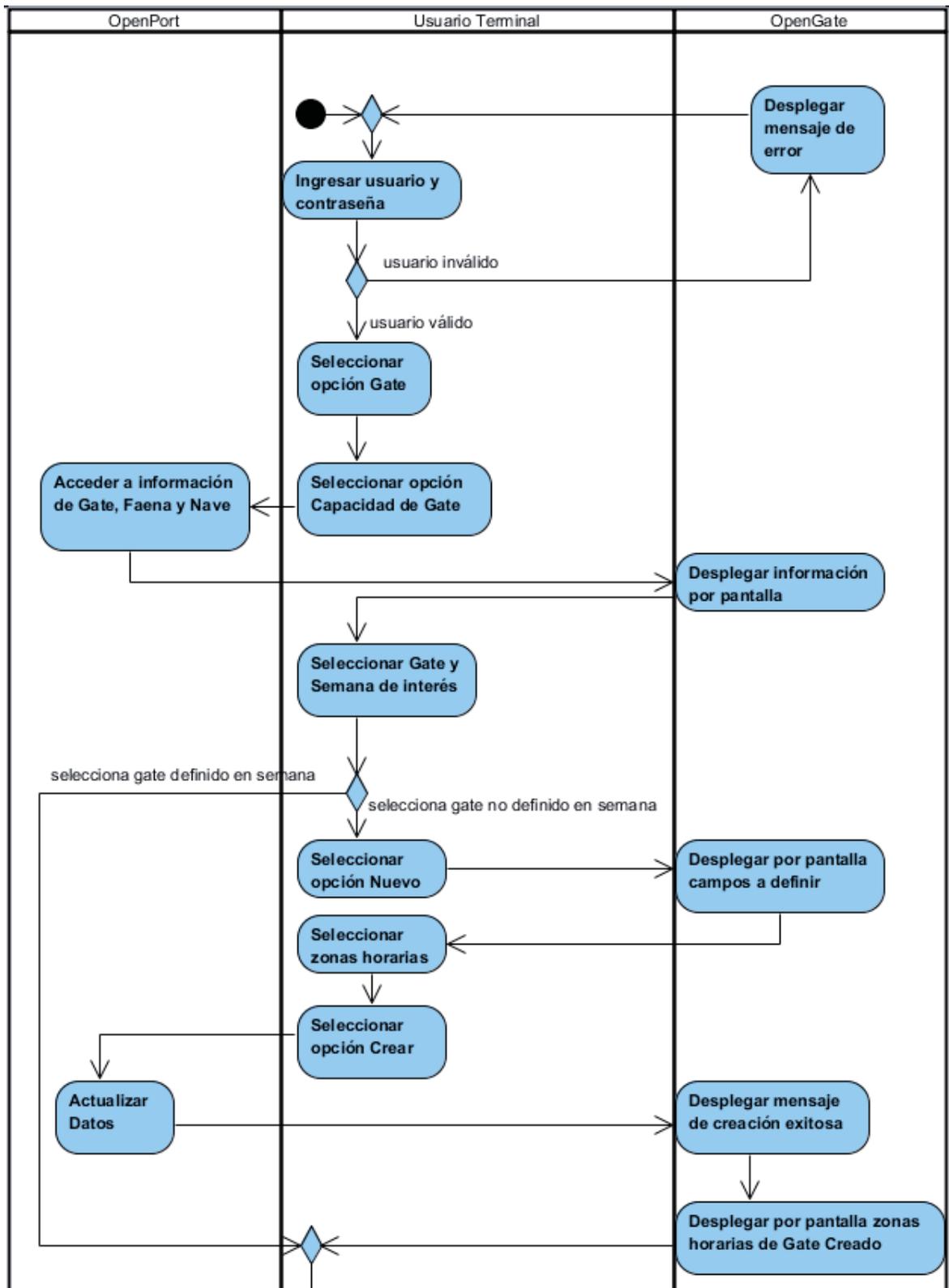
El Diagrama de actividades es un modelo mayoritariamente dinámico, que muestra y organiza temporalmente y describe las posibilidades procedimentales del sistema modelado sobre la base de actividades (carácter funcional).

Para la construcción del Diagrama de actividades se presenta la Tabla 4.16 que corresponde a las herramientas disponibles y sus usos. Antes de presentar los diagramas es necesario decir que para efectos de mayor entendimiento del sistema y mostrar la esencia del sistema OpenGate, se optó por omitir la excepción general *a de la documentación de casos de uso (sección 4.2.2) que consiste en terminar los CU si la conexión a internet se ve interrumpida, esto para evitar caer en perder el objetivo de mostrar el comportamiento y secuenciamiento del sistema.

Tabla 4.16 - Herramientas y su utilización para construcción de DAct.

Herramienta	Utilización
Documentación de Casos de Uso (DoCU).	Comportamiento y Secuenciación.
Prototipo OpenGate (www.corleone.cl/opengate).	Nodo final y Nodos de términos de flujo.

A continuación se muestran los Diagramas de Actividades (en el mismo orden que la DoCU sección 4.2.2) correspondientes a detallar cómo se comporta el sistema en cada Caso de Uso definido en la sección 4.2.1.



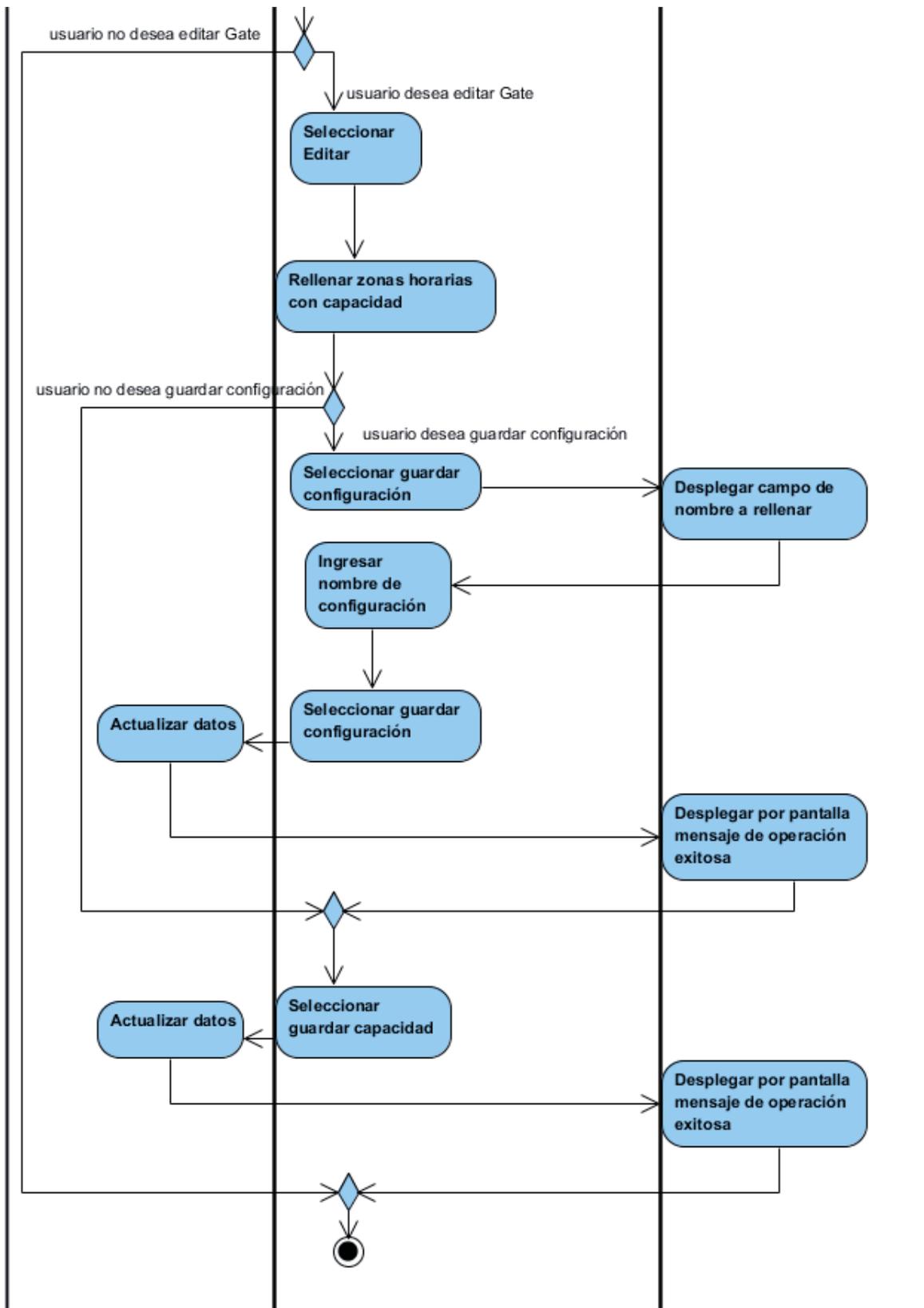
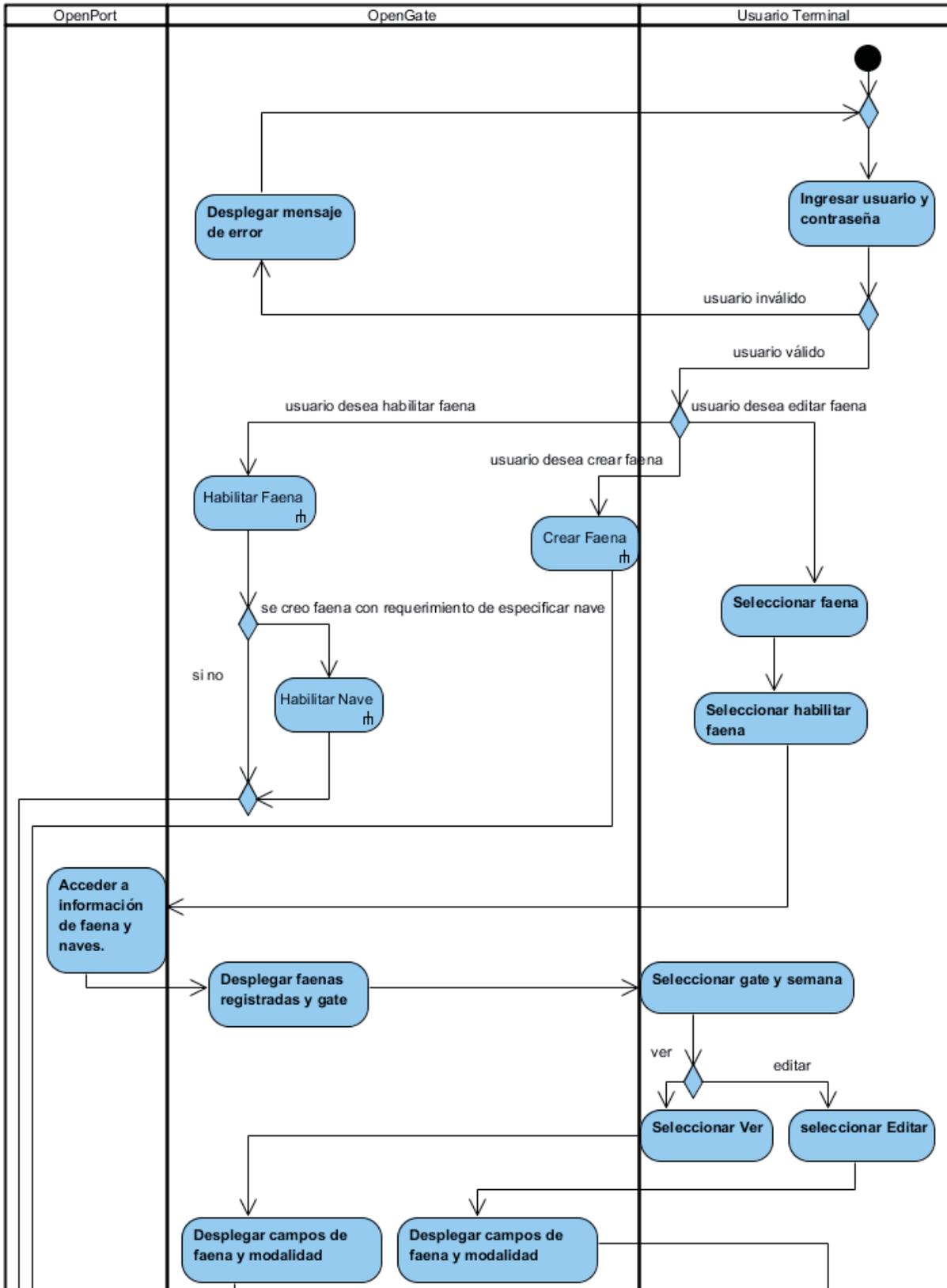


Figura 4.5 – Diagrama de Actividades Administrar Capacidad Gate.



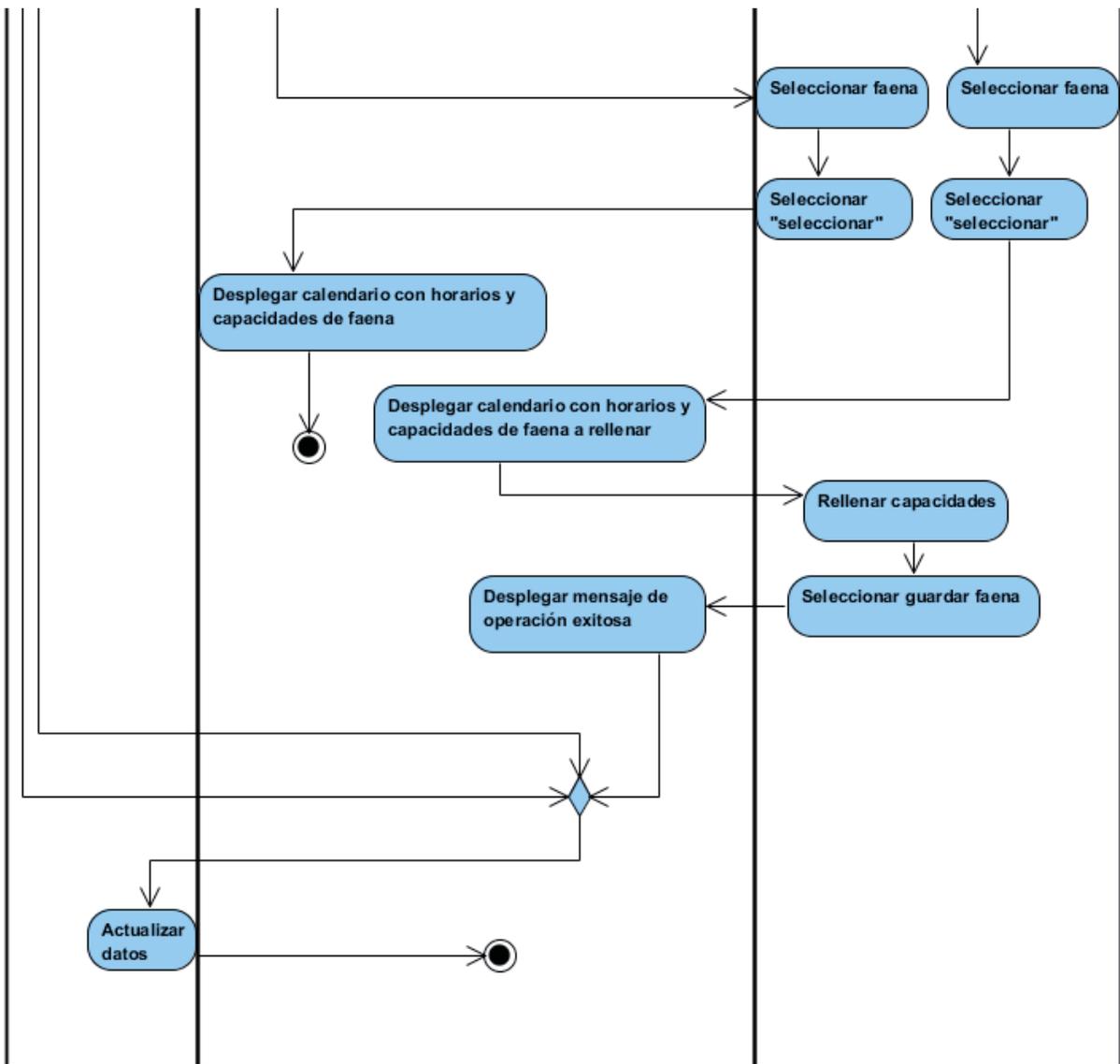


Figura 4.6 – Diagrama de Actividades Administrar Faenas.

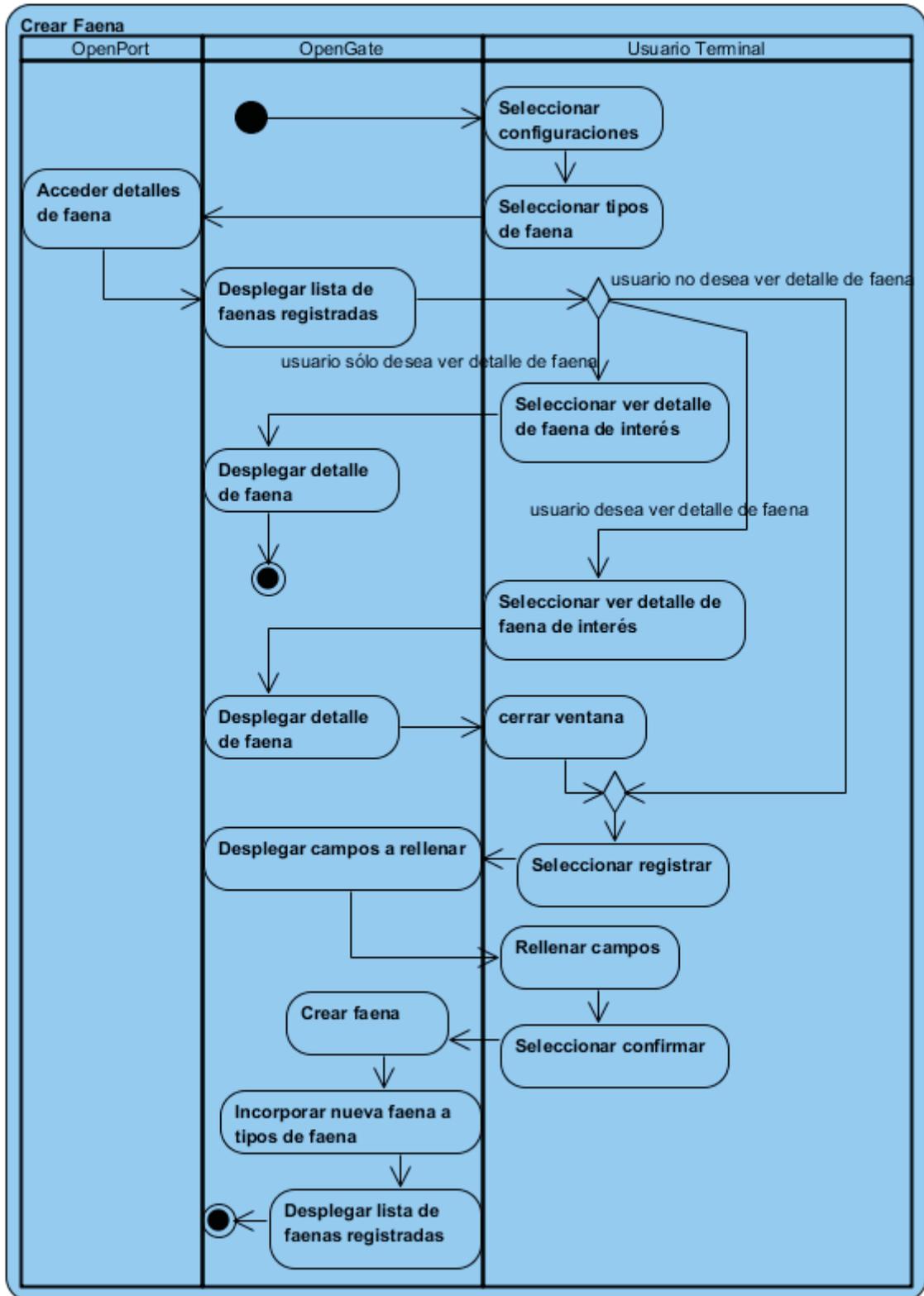


Figura 4.7 – Diagrama de Actividades Crear Faenas.

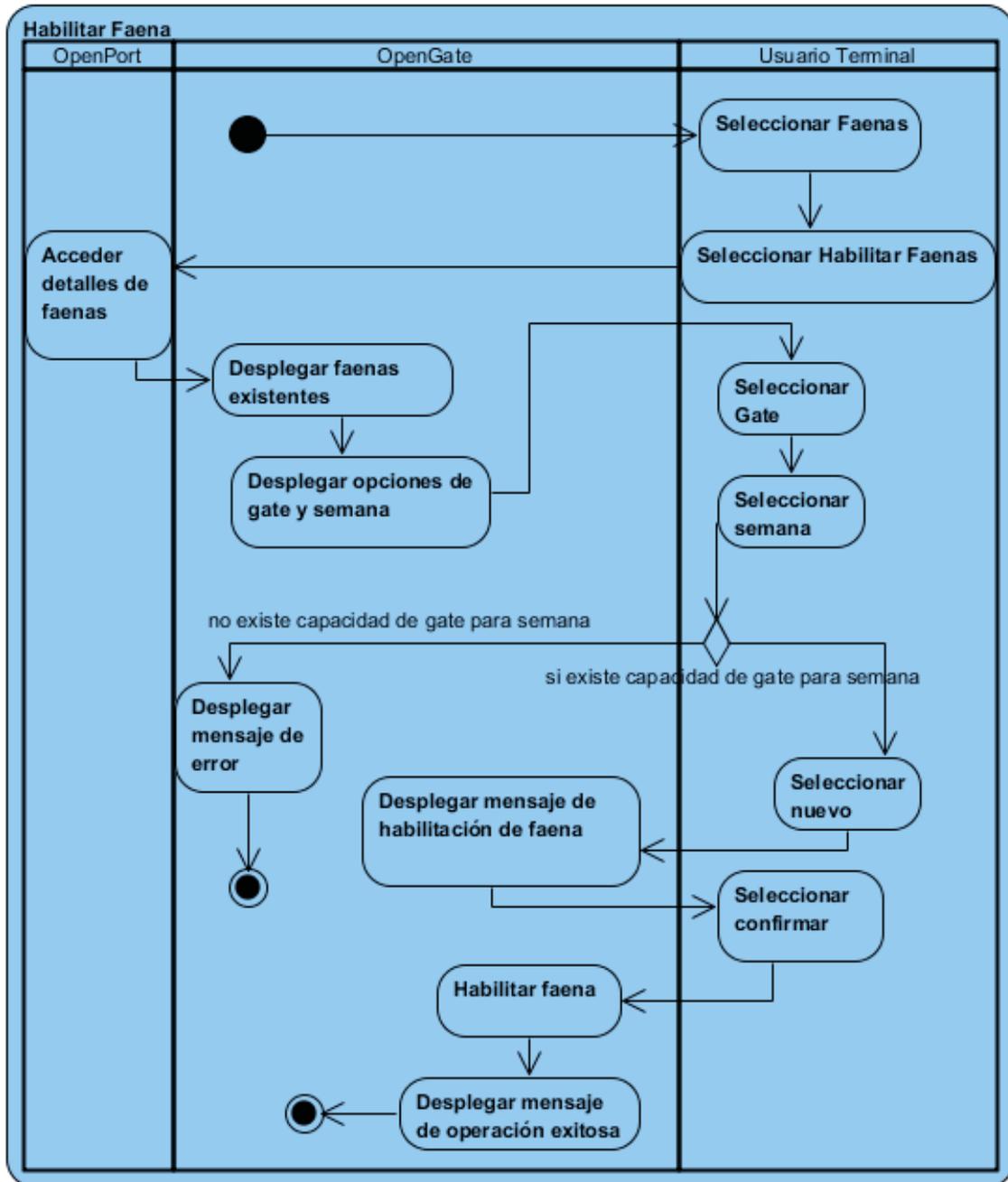
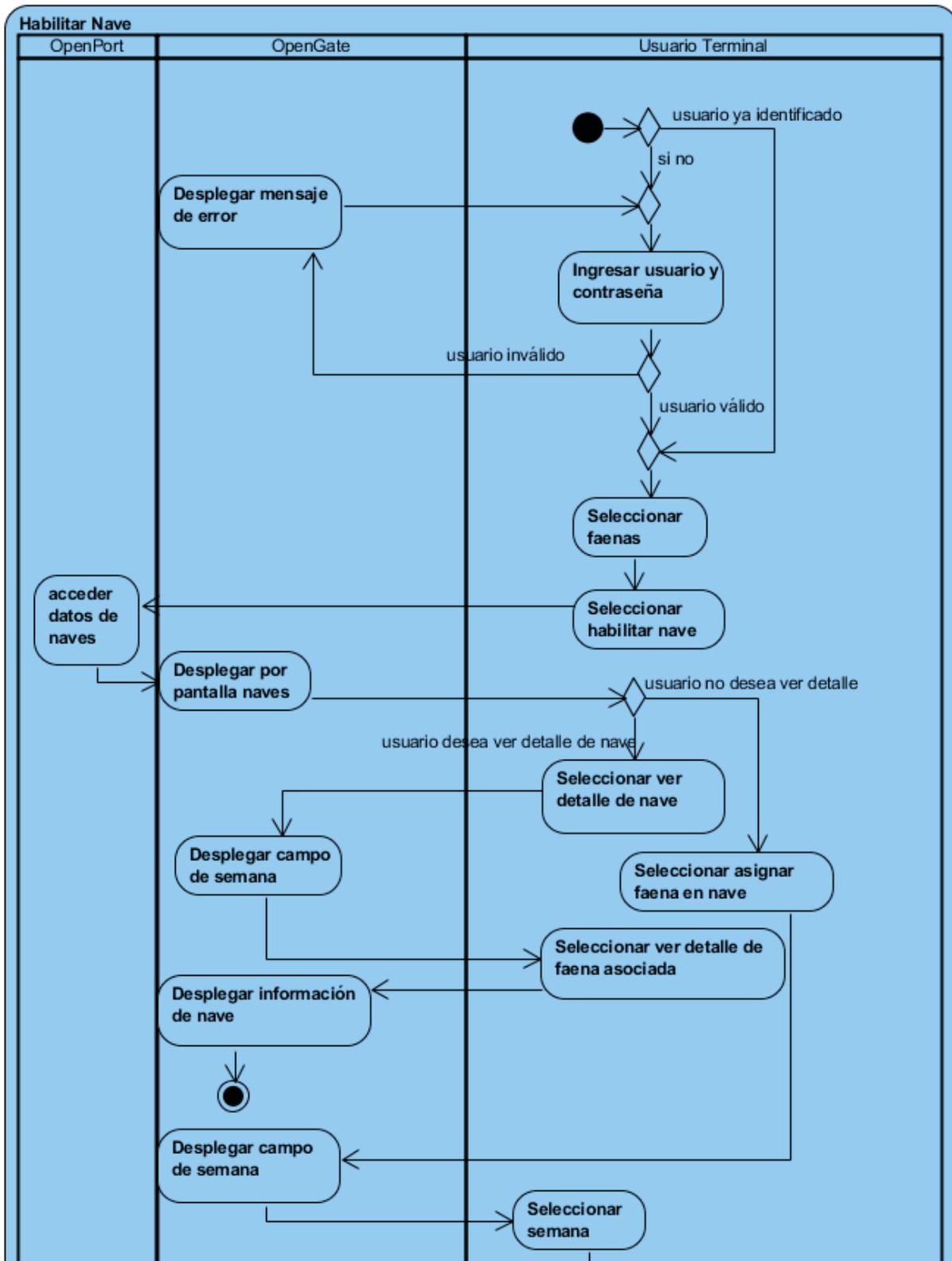


Figura 4.8 – Diagrama de Actividades Habilitar Faenas.



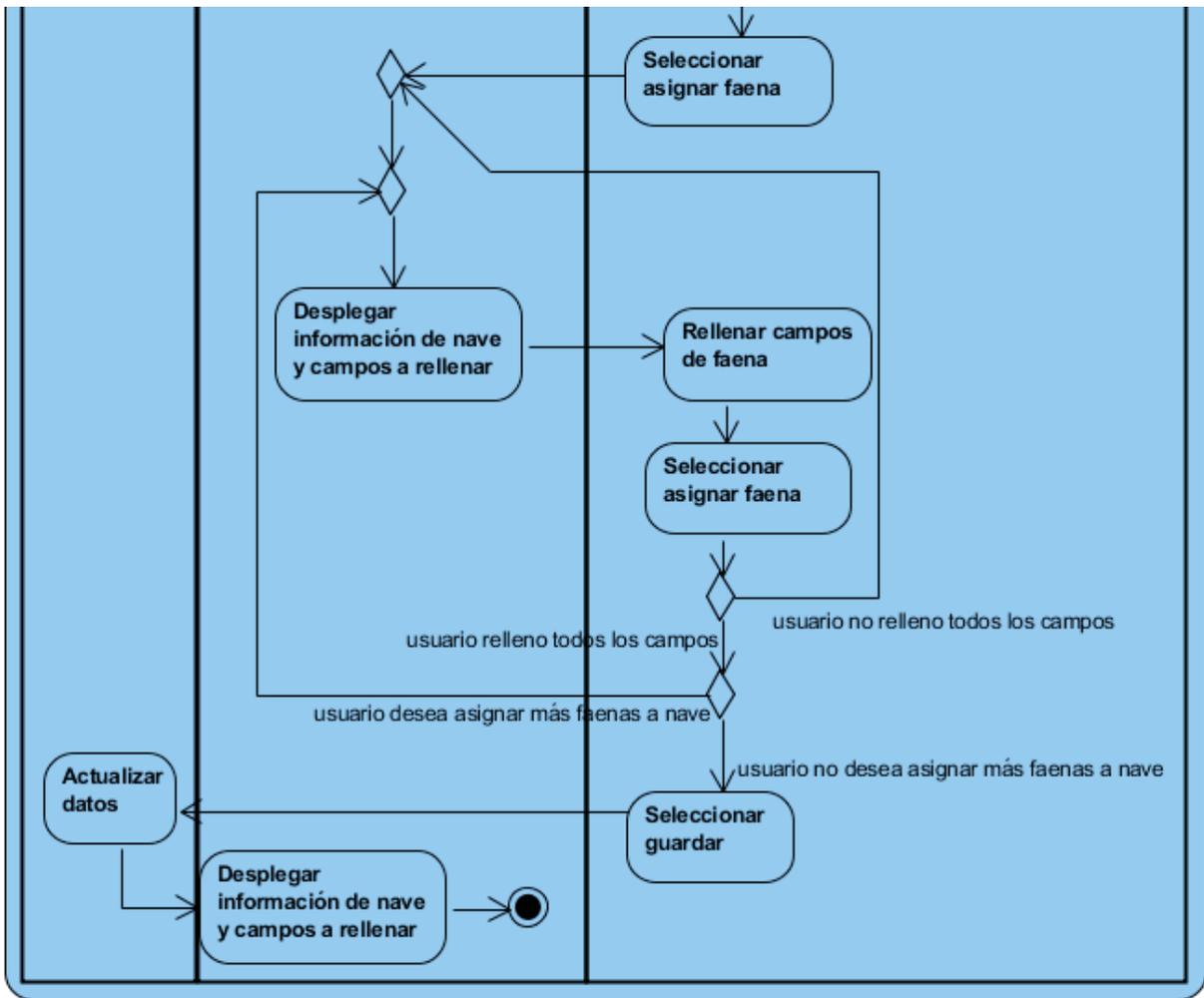
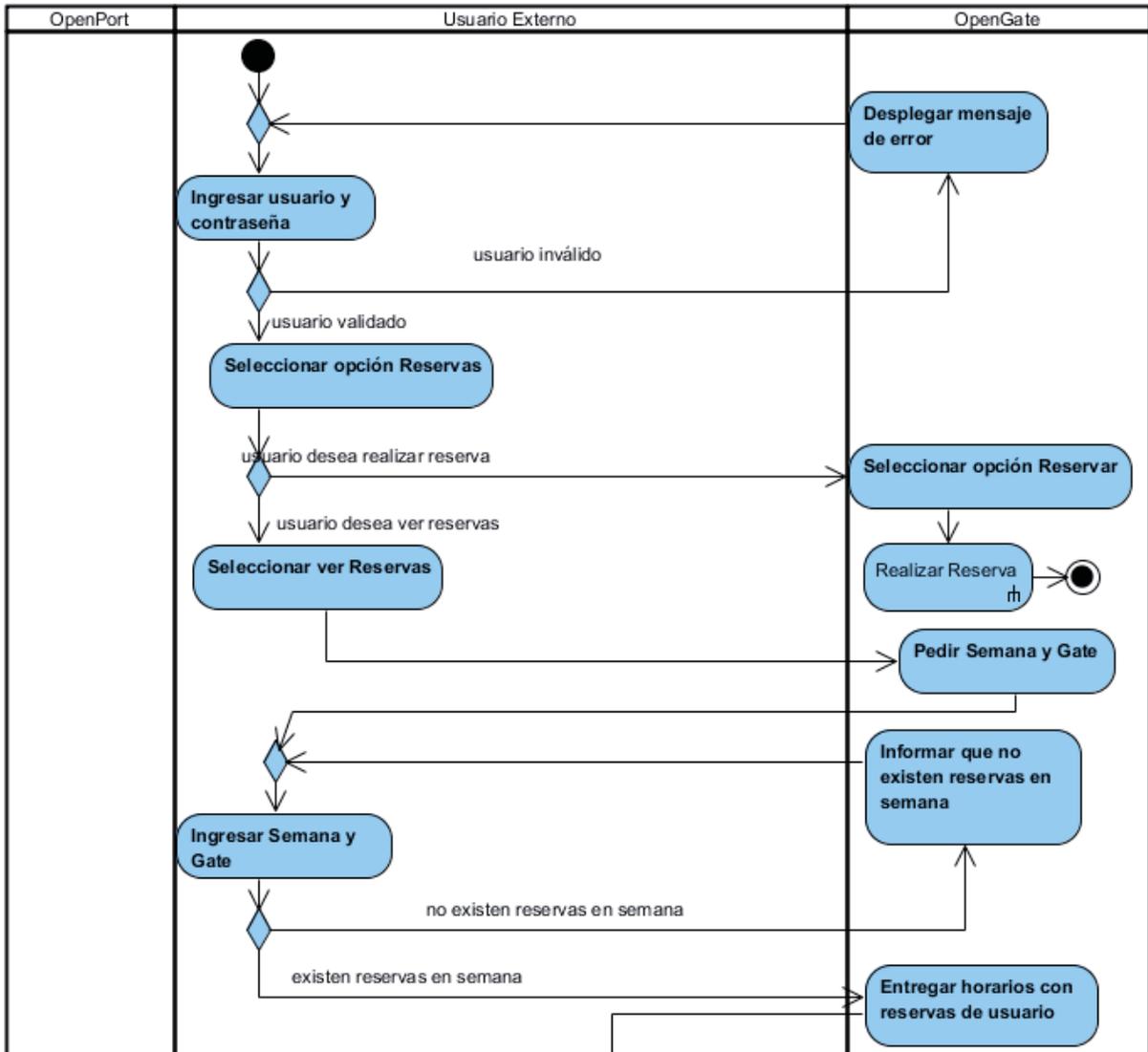


Figura 4.9 – Diagrama de Actividades Habilitar Naves.



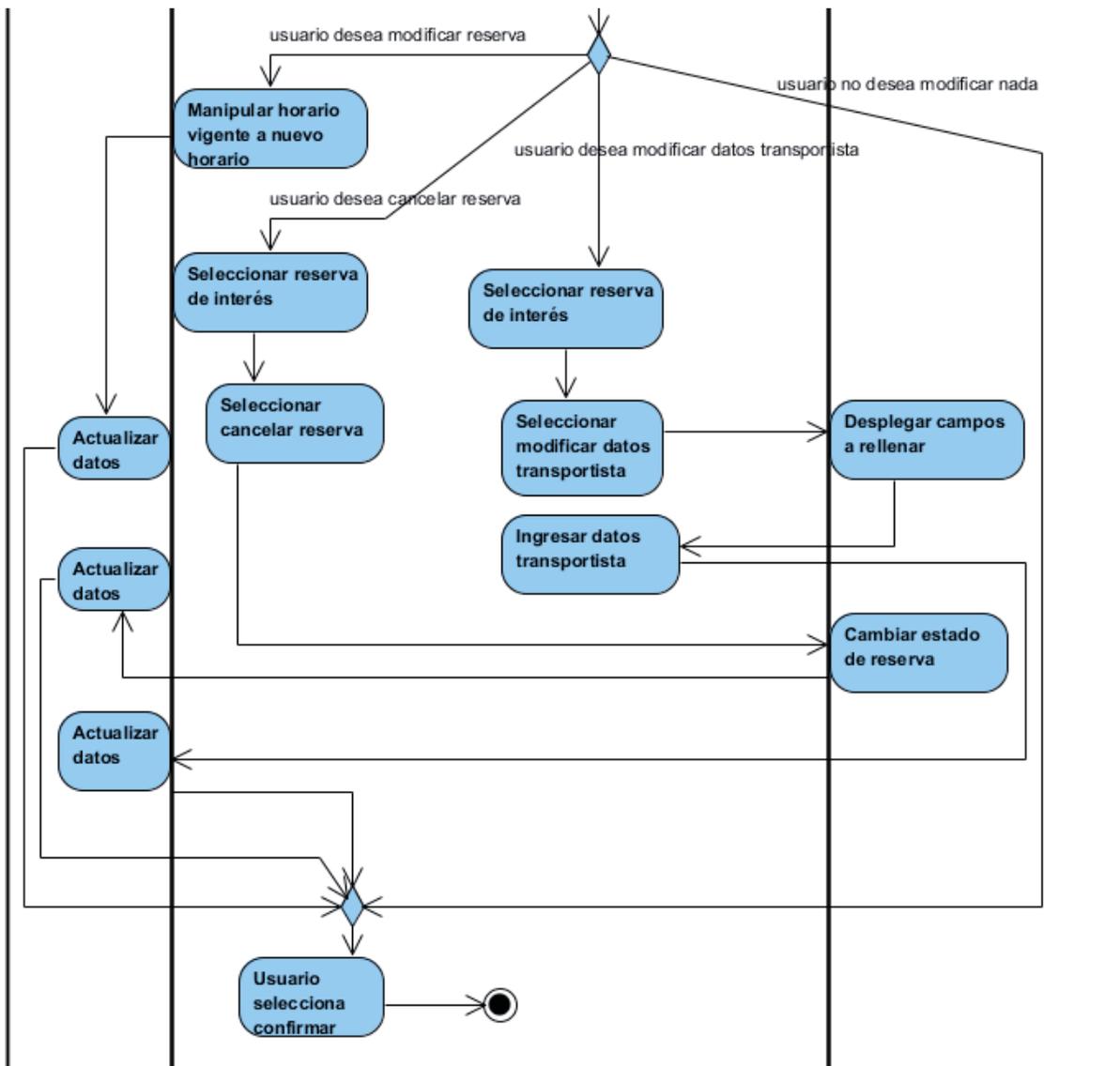
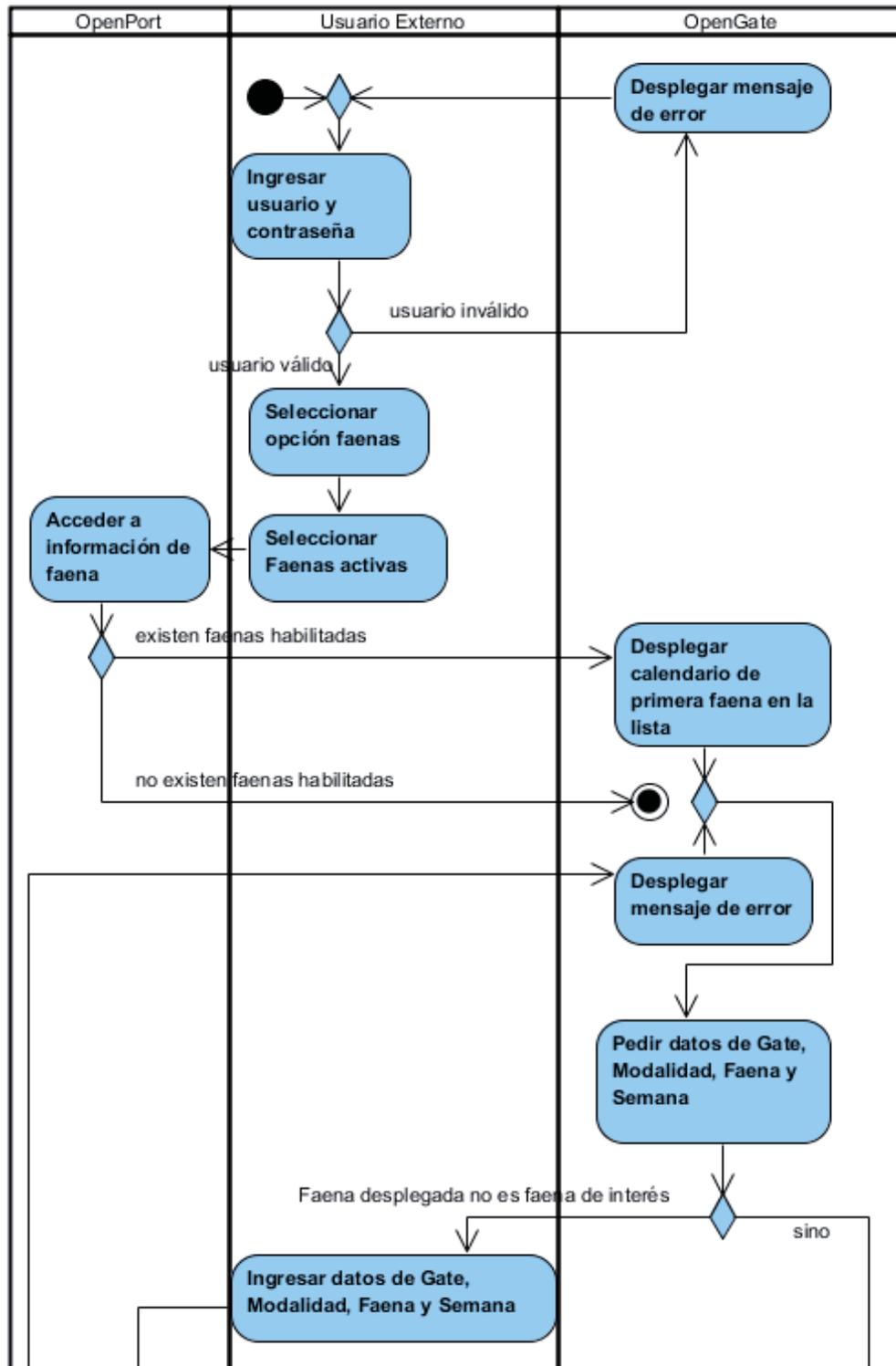


Figura 4.10 – Diagrama de Actividades Administrar Reservas.



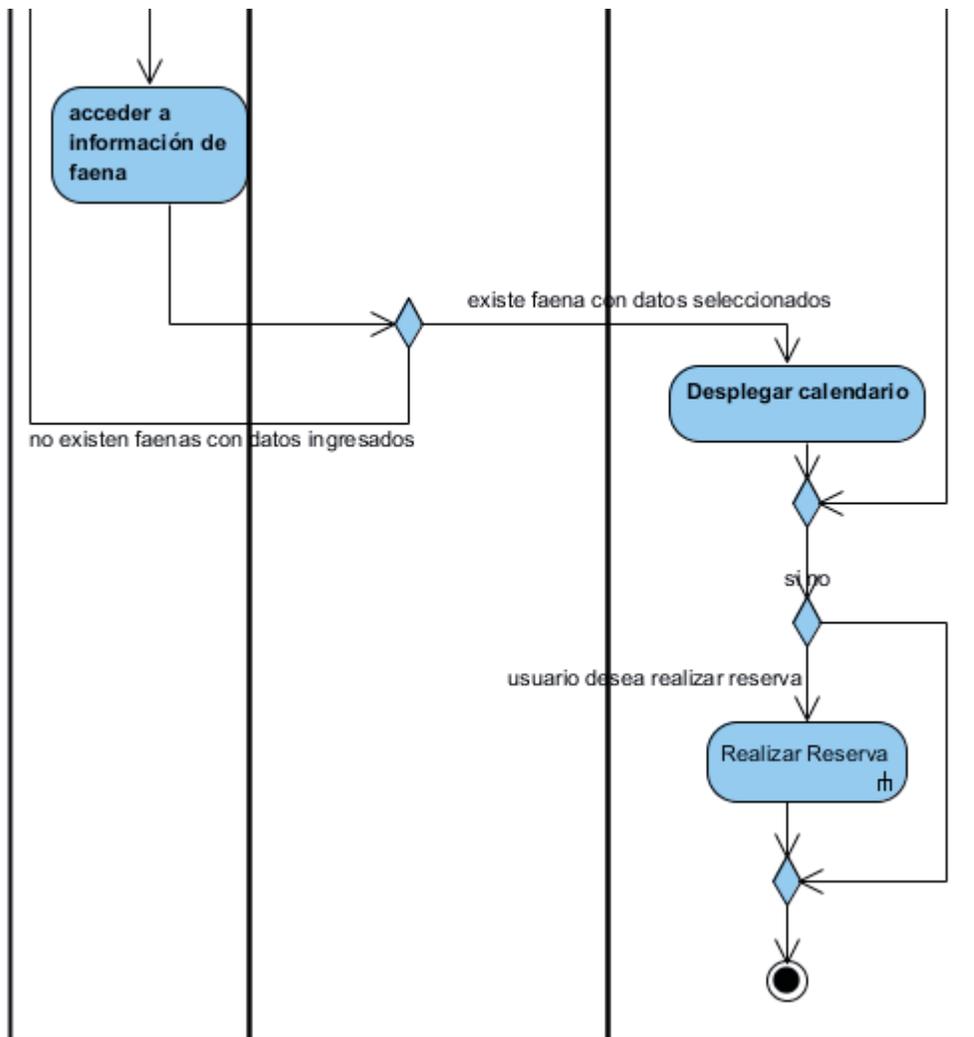


Figura 4.11 – Diagrama de Actividades Ver Faenas Activas.

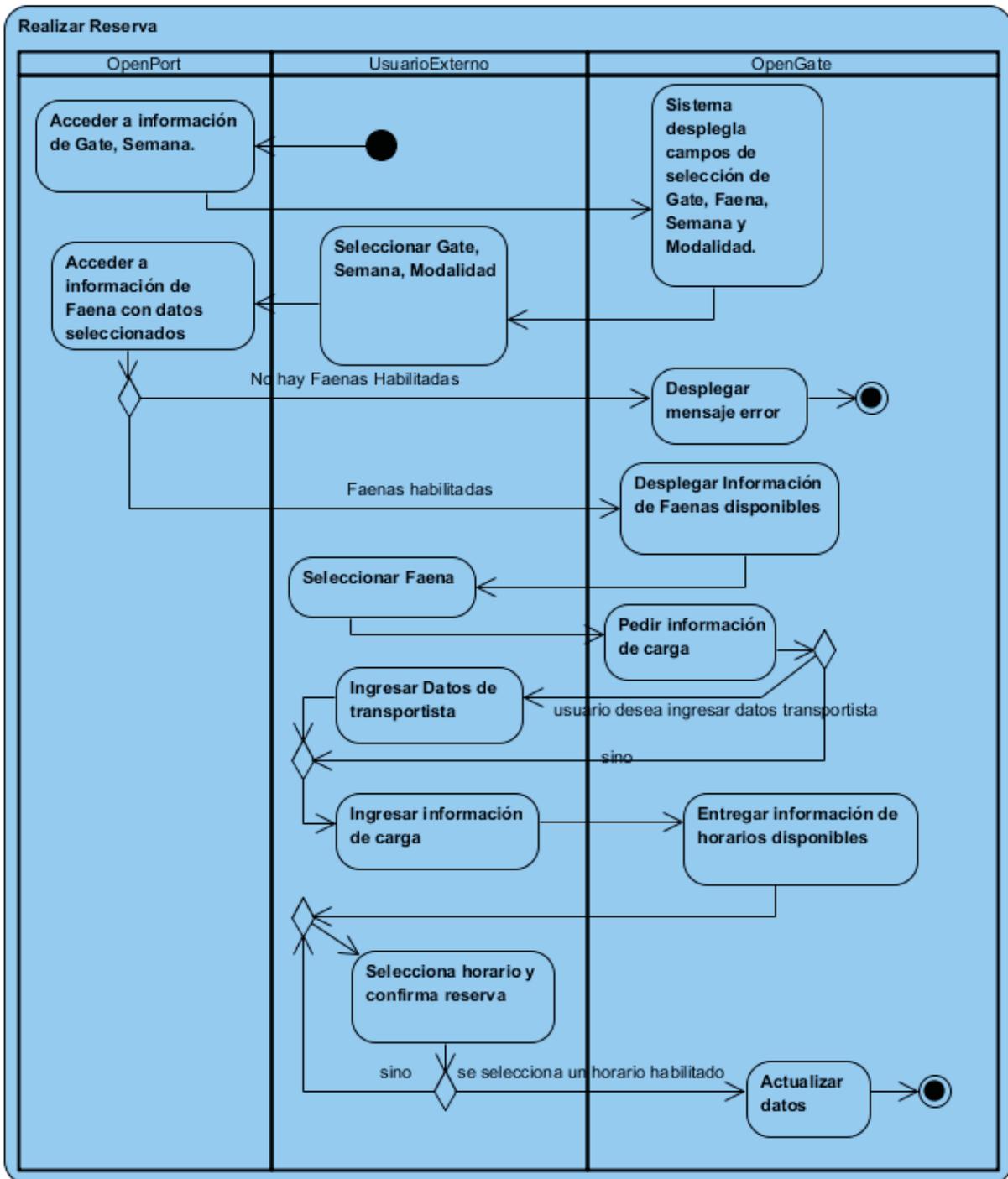
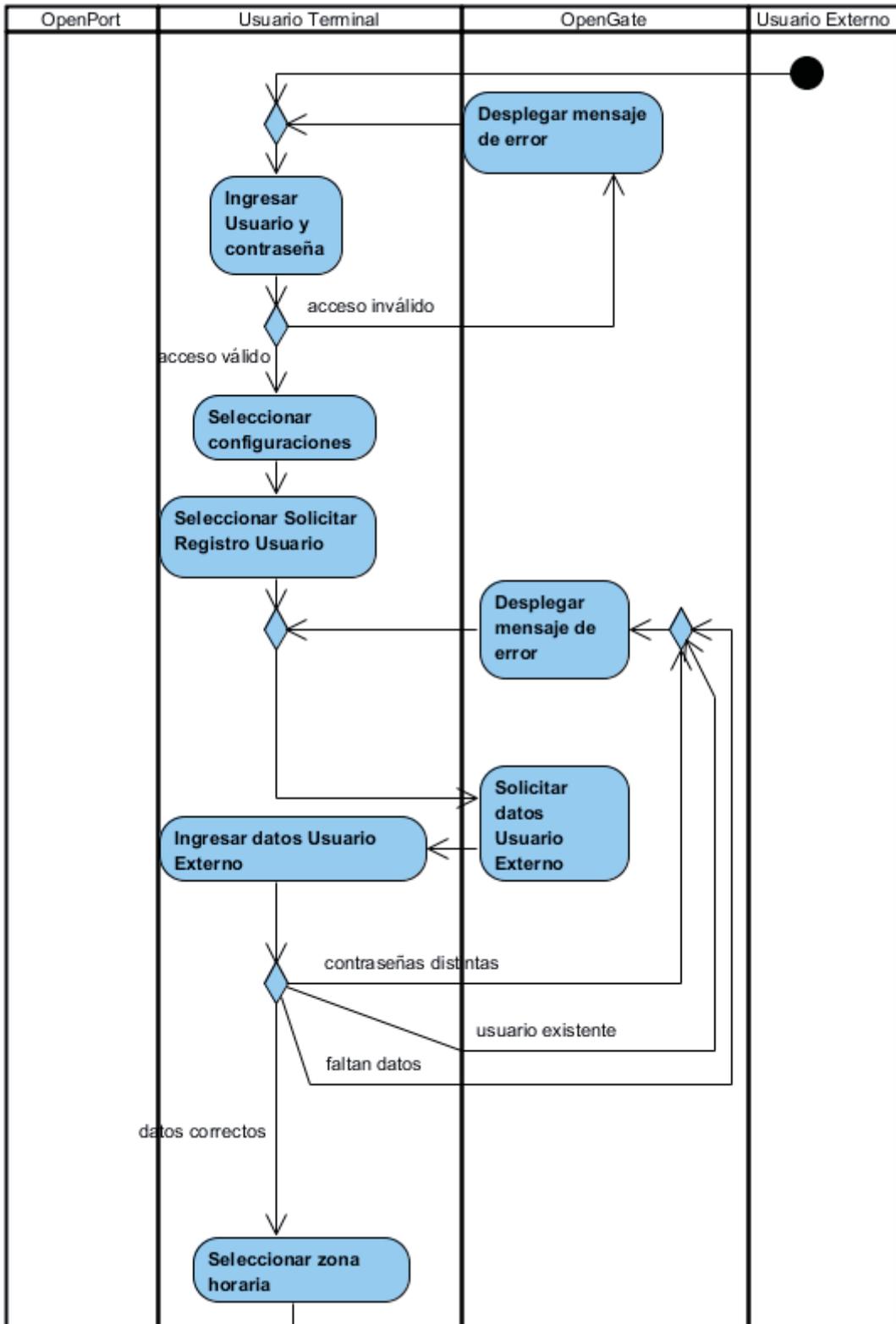


Figura 4.12 – Diagrama de Actividades Realizar Reserva.



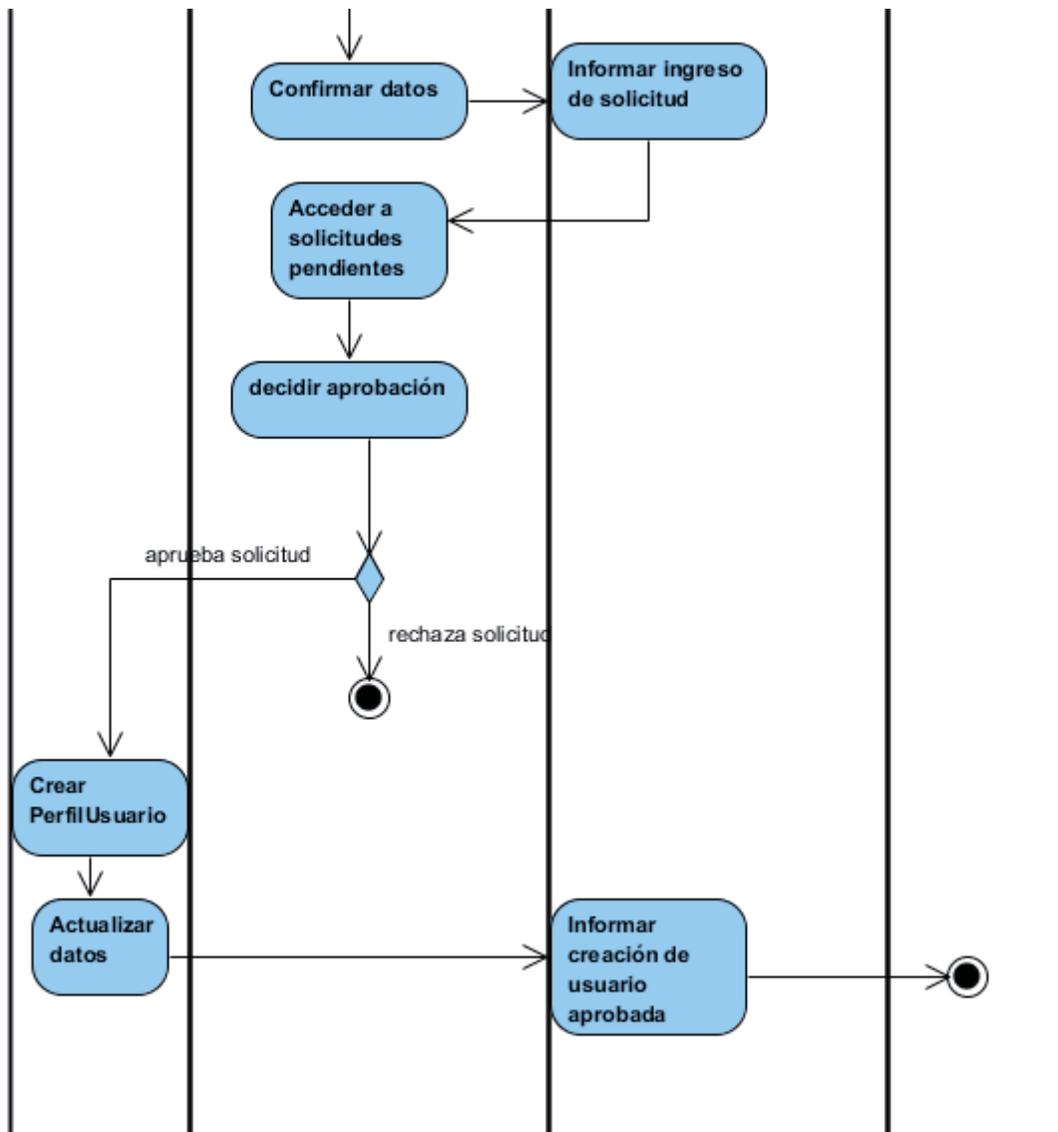


Figura 4.13 – Diagrama de Actividades Crear Usuario Externo.

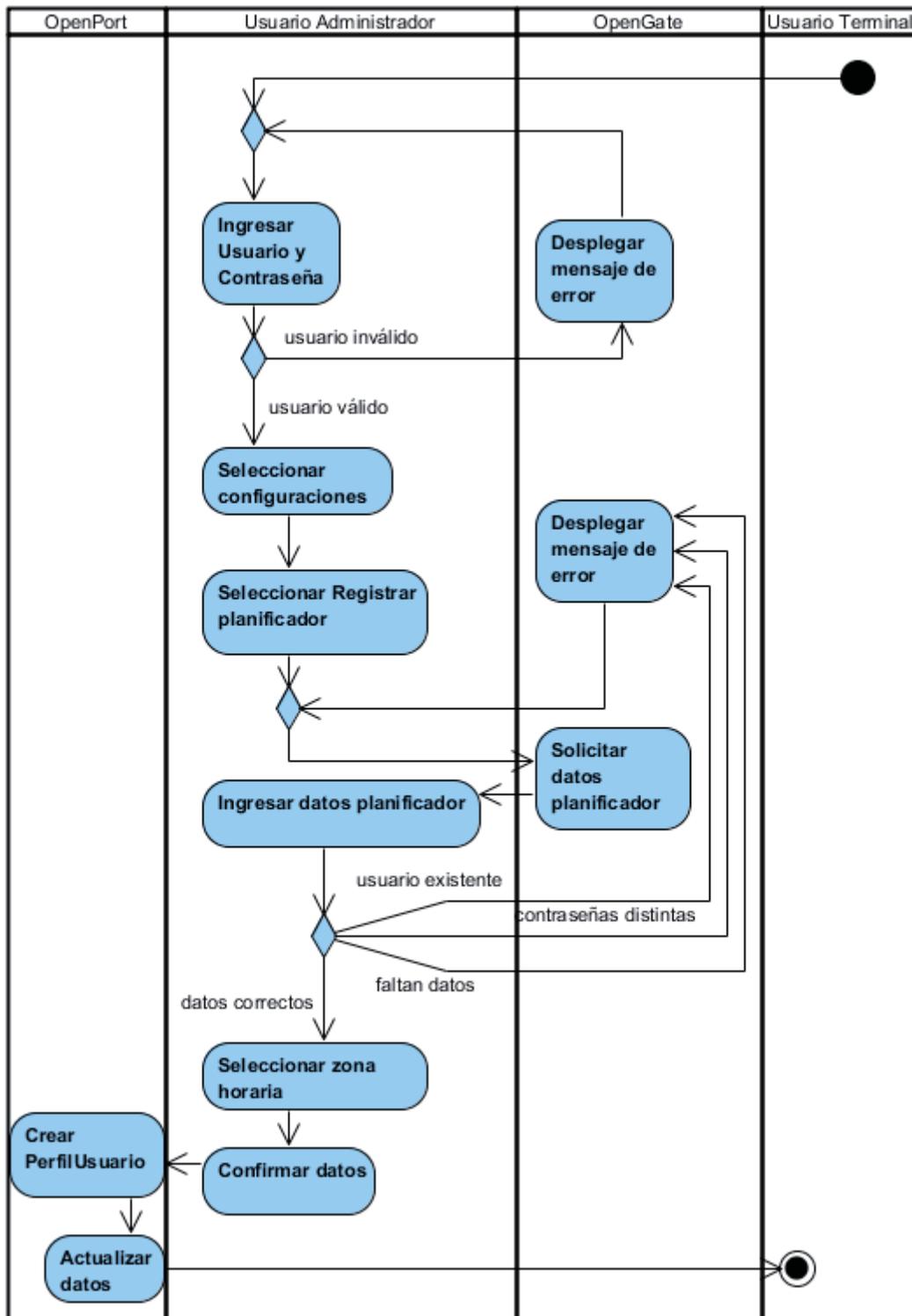


Figura 4.14 – Diagrama de Actividades Crear Usuario Terminal.

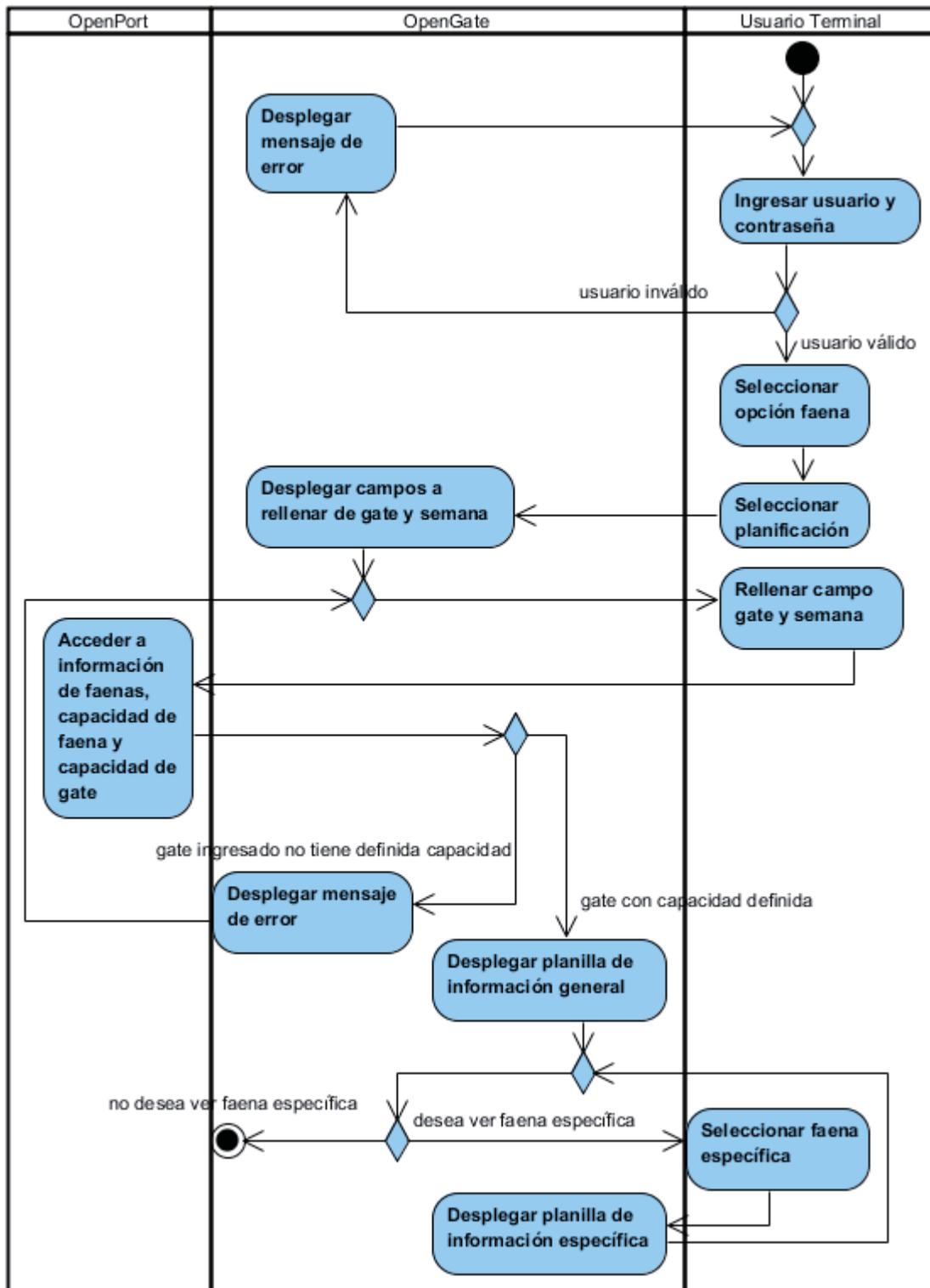
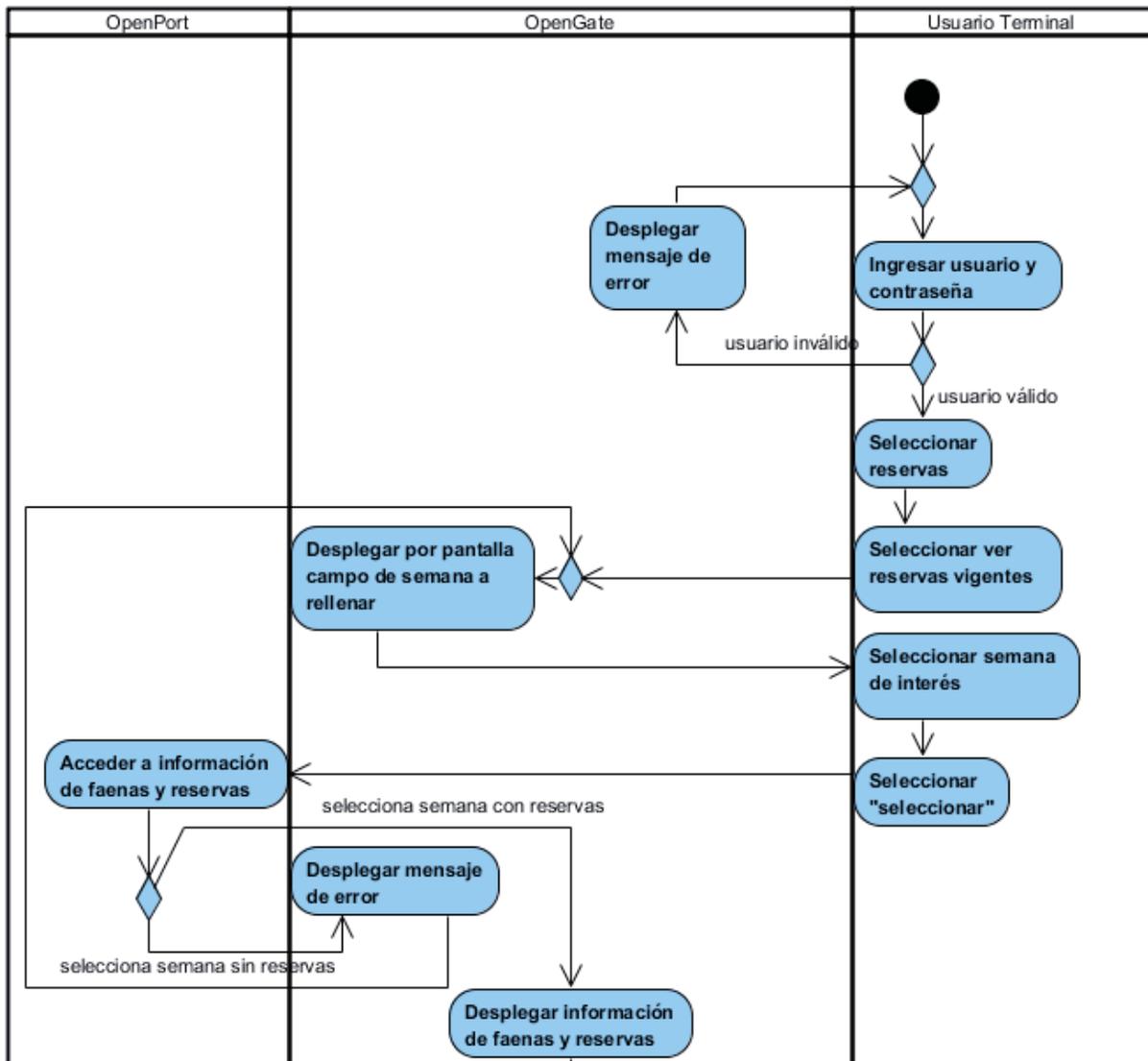


Figura 4.15 – Diagrama de Actividades Ver Planificación.



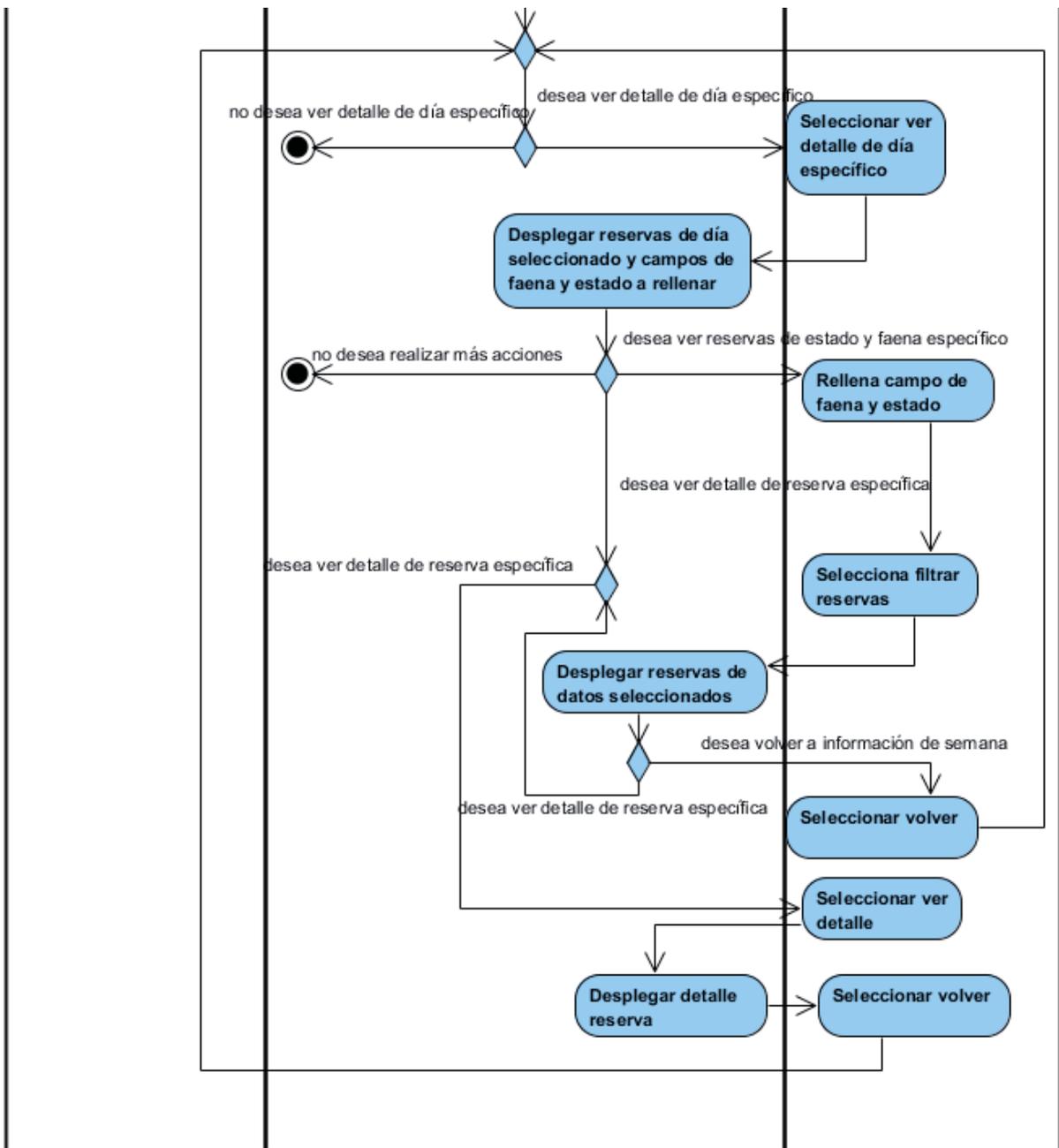


Figura 4.16 – Diagrama de Actividades Ver Reservas Vigentes.

4.3.2 Diagramas de Máquinas de Estado (DME)

El DME es el modelo más elaborado para la dimensión dinámica en términos de comportamiento, este muestra cómo se comporta el sistema frente a estímulos o mensajes recibidos, es decir, representa los cambios de estados que sufre el sistema en respuesta a una acción anterior.

Cada estructura en el DCla corresponde un DME asociado, en donde se ven los cambios de estado de la clase frente a la invocación o recibimiento de alguna operación en el sistema, ahora se muestra la Tabla 4.17, en donde se vislumbra las herramientas y su utilización para la construcción de los modelos, para luego después presentar los DME asociados a cada clase en el DCla.

Tabla 4.17 - Herramientas y su utilización para construcción de DME.

Herramienta	Utilización
Dsec	Métodos (mensajes).
DCla	Clases y operaciones.

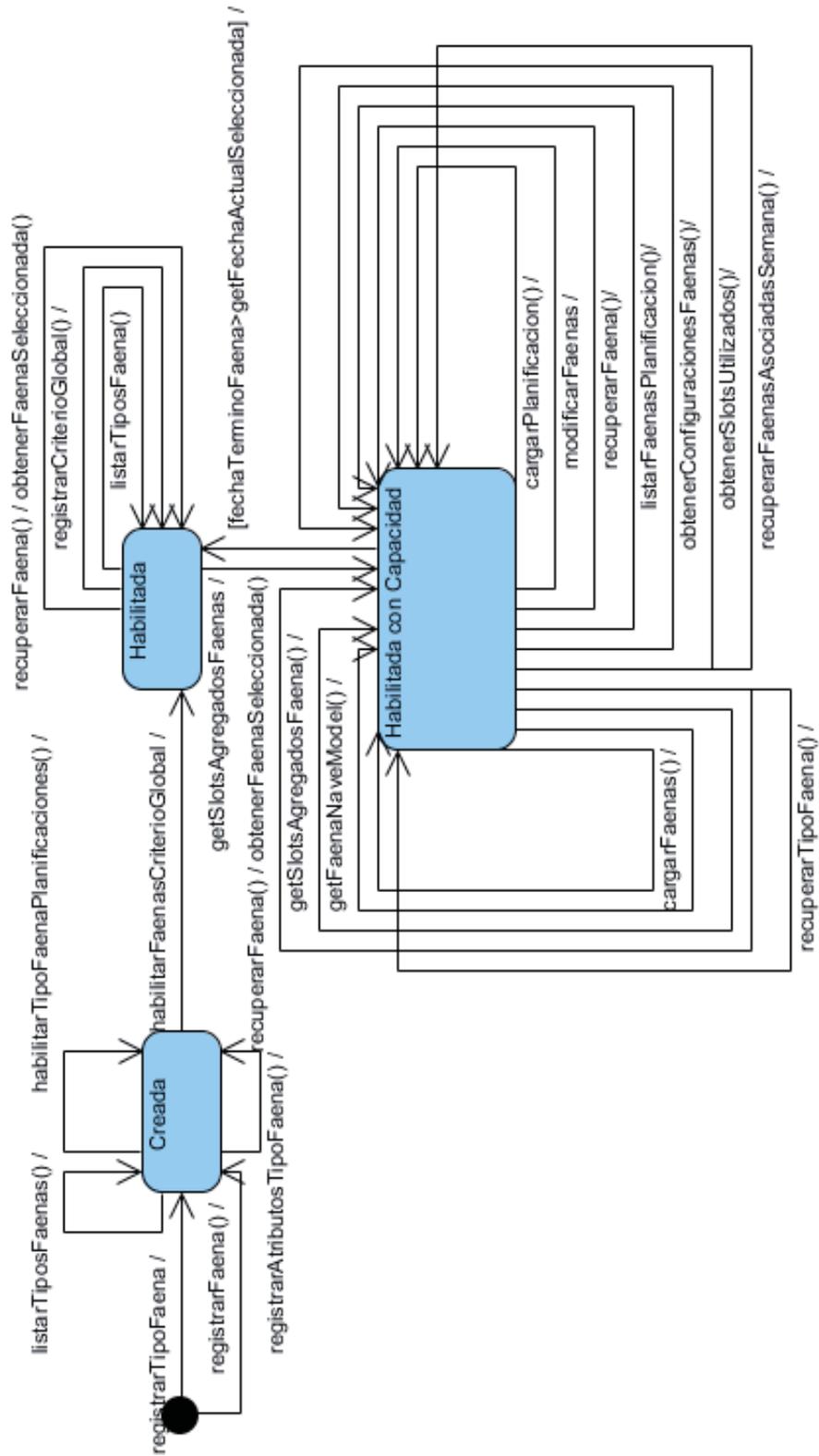


Figura 4.17 – DME Faenas.

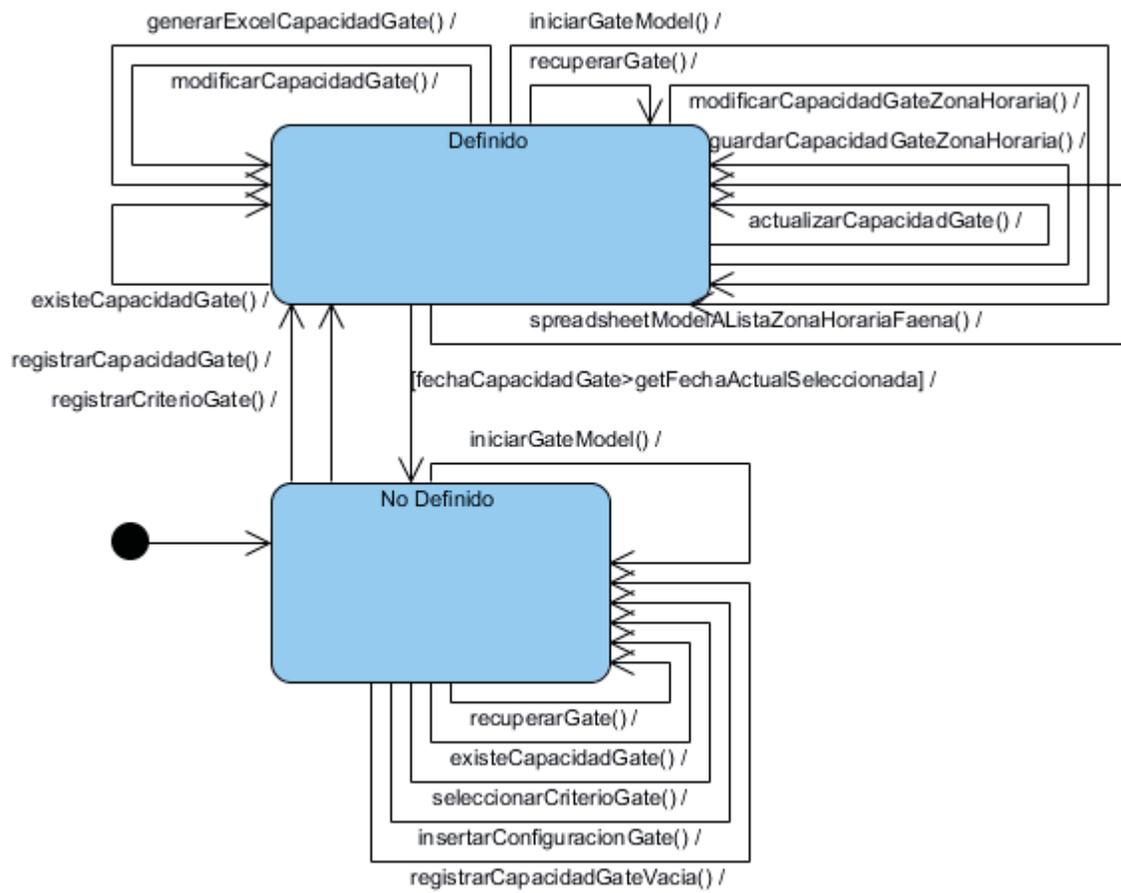


Figura 4.18 – DME Gate.

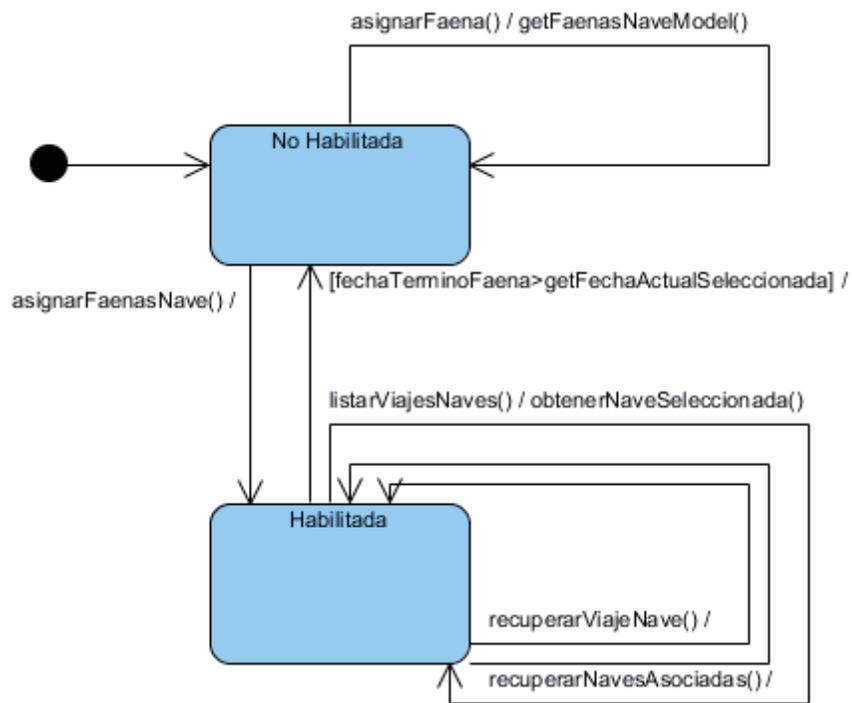


Figura 4.19 – DME Nave.

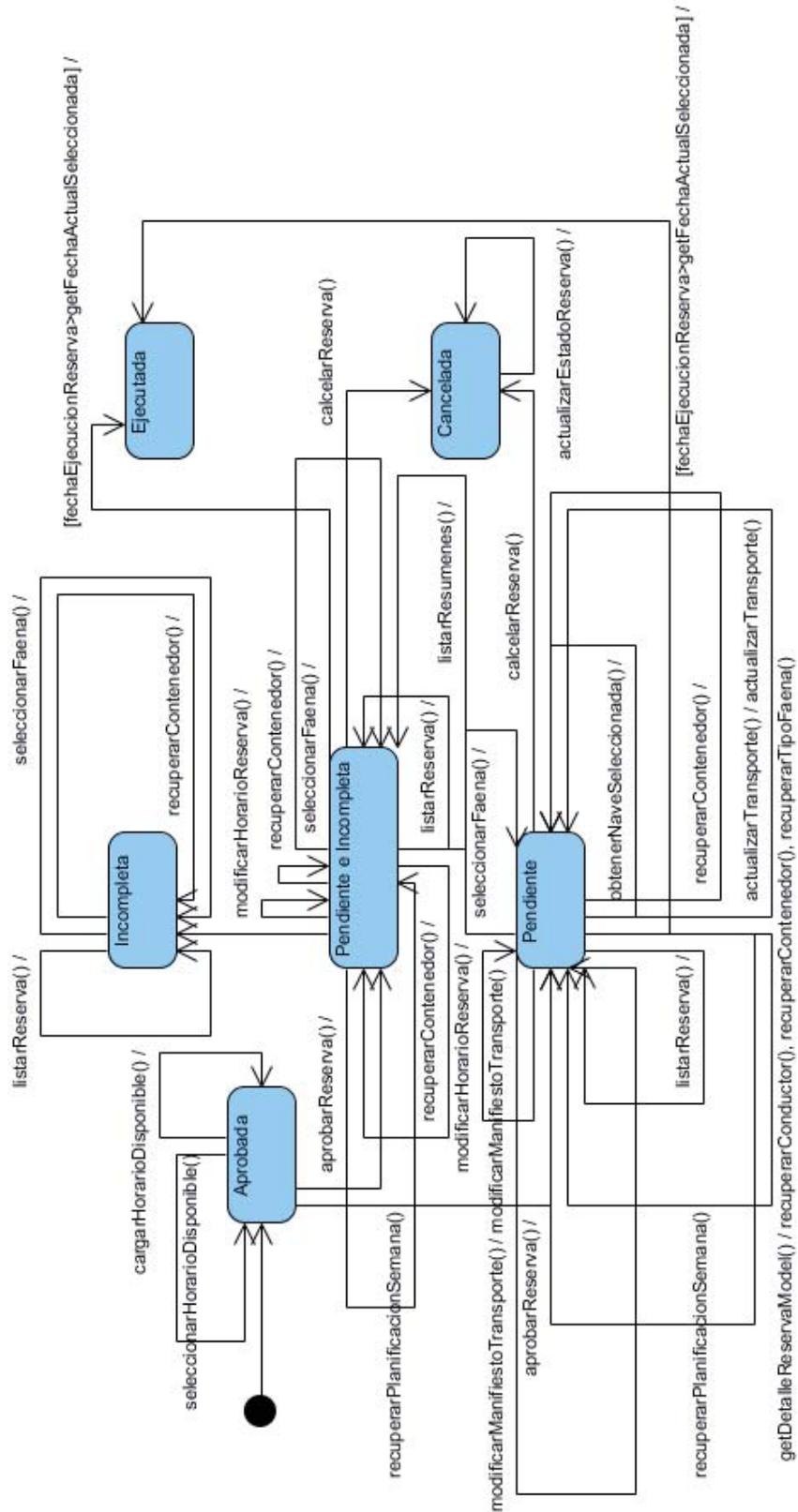


Figura 4.20 – DME Reserva.

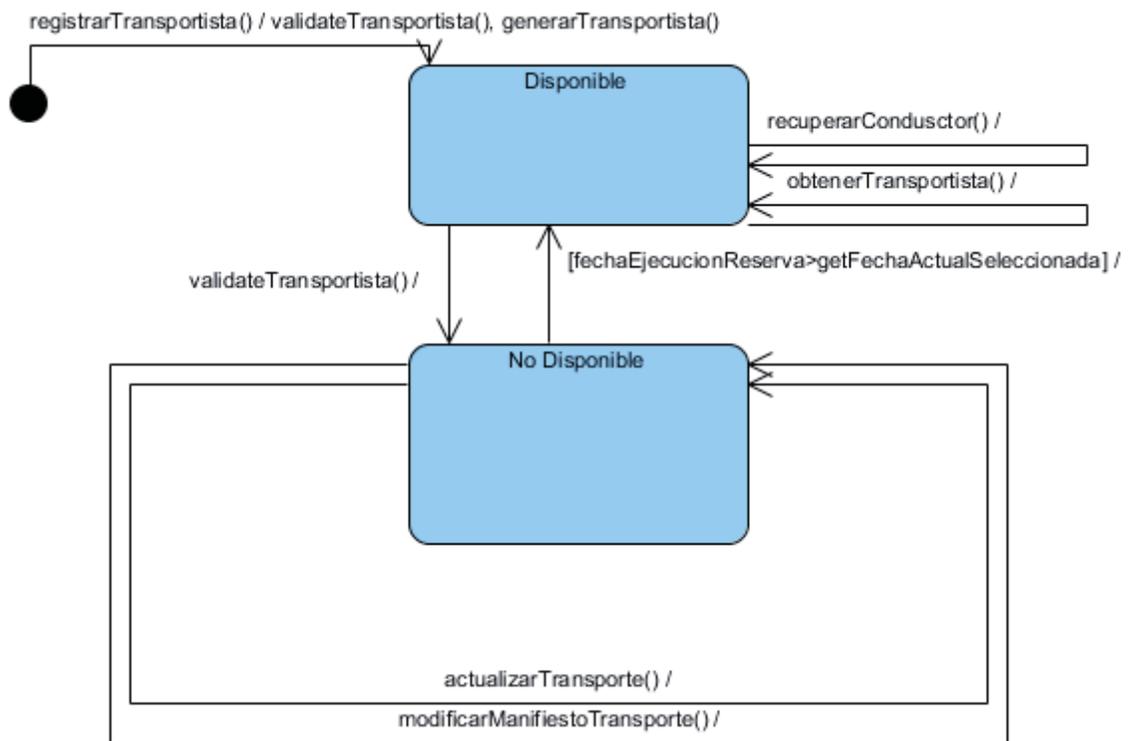


Figura 4.21 – DME Transporte.

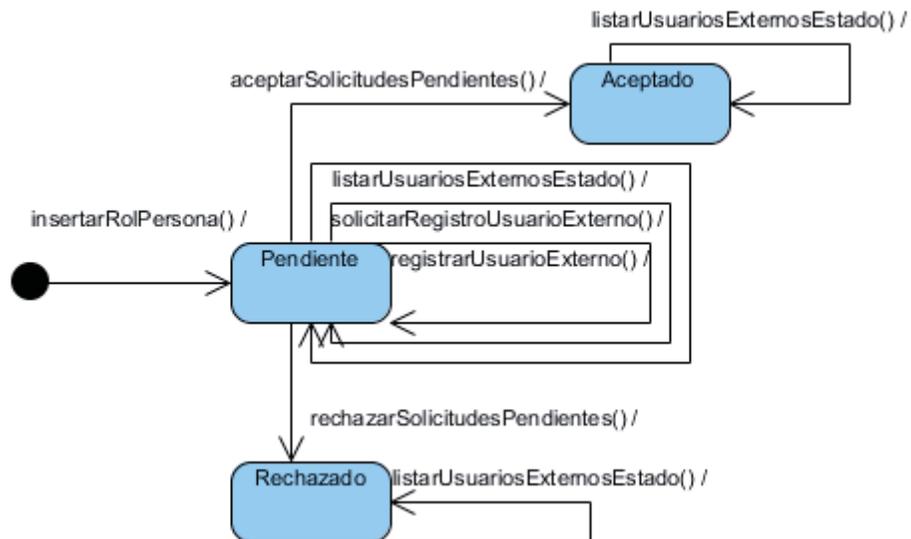


Figura 4.22 – DME UsuarioExterno.

4.3.3 Diagramas de Secuencia (DSec)

El DSec es un modelo que muestra el conjunto de interacciones entre componentes en un sistema, en forma cronológica, para un contexto temporal delimitado, ordena los mensajes que se emiten a partir de un estímulo o evento recibido por el sistema. En la integración UML cada Caso de Uso debe tener un subdiagrama DSec asociado que muestra la interacción de los componentes definidos en el DCIa para cumplir la funcionalidad que el Caso de Uso propone. Cabe mencionar que los mensajes serán encapsulados de manera que los diagramas sean más simples de entender, explicado en la sección 4 (supuestos adoptados para la documentación).

Como también se explica en la sección del punto 4, se omitirán todos los registros y validaciones de usuarios correspondientes a la clase `OpenGateUserDetails` perteneciente al controlador, se aprecia en la figura 4.23.

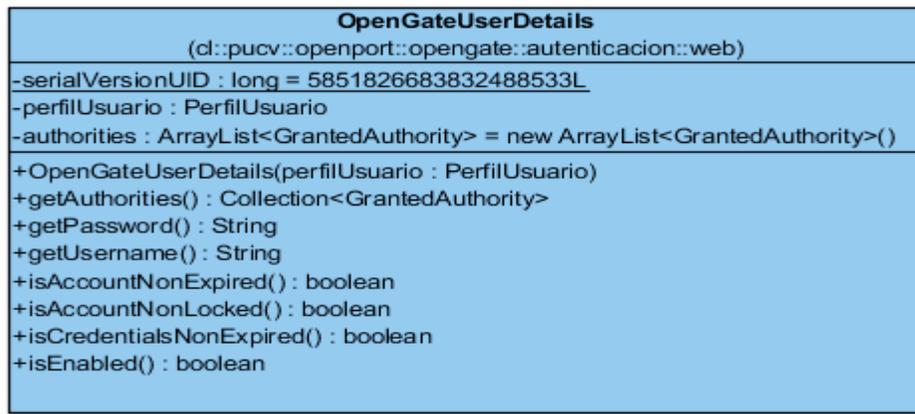


Figura 4.23 – Clase `OpenGateUserDetails`. (Fuente: OpenGate, 2014).

A continuación se muestra la Tabla 4.18 en donde se vislumbran las herramientas y su utilización en la construcción de los diagramas, para luego mostrar los DSec correspondientes a cada Caso de Uso mostrado en la sección 4.2.1.

Tabla 4.18 - Herramientas y su utilización para construcción de DSec.

Herramienta	Utilización
DAct	Orden de actividades.
Código Java	Métodos (mensajes).
Prototipo (www.corleone.cl/opengate)	Orden de Mensajes
DCIa	Componentes del Diagrama

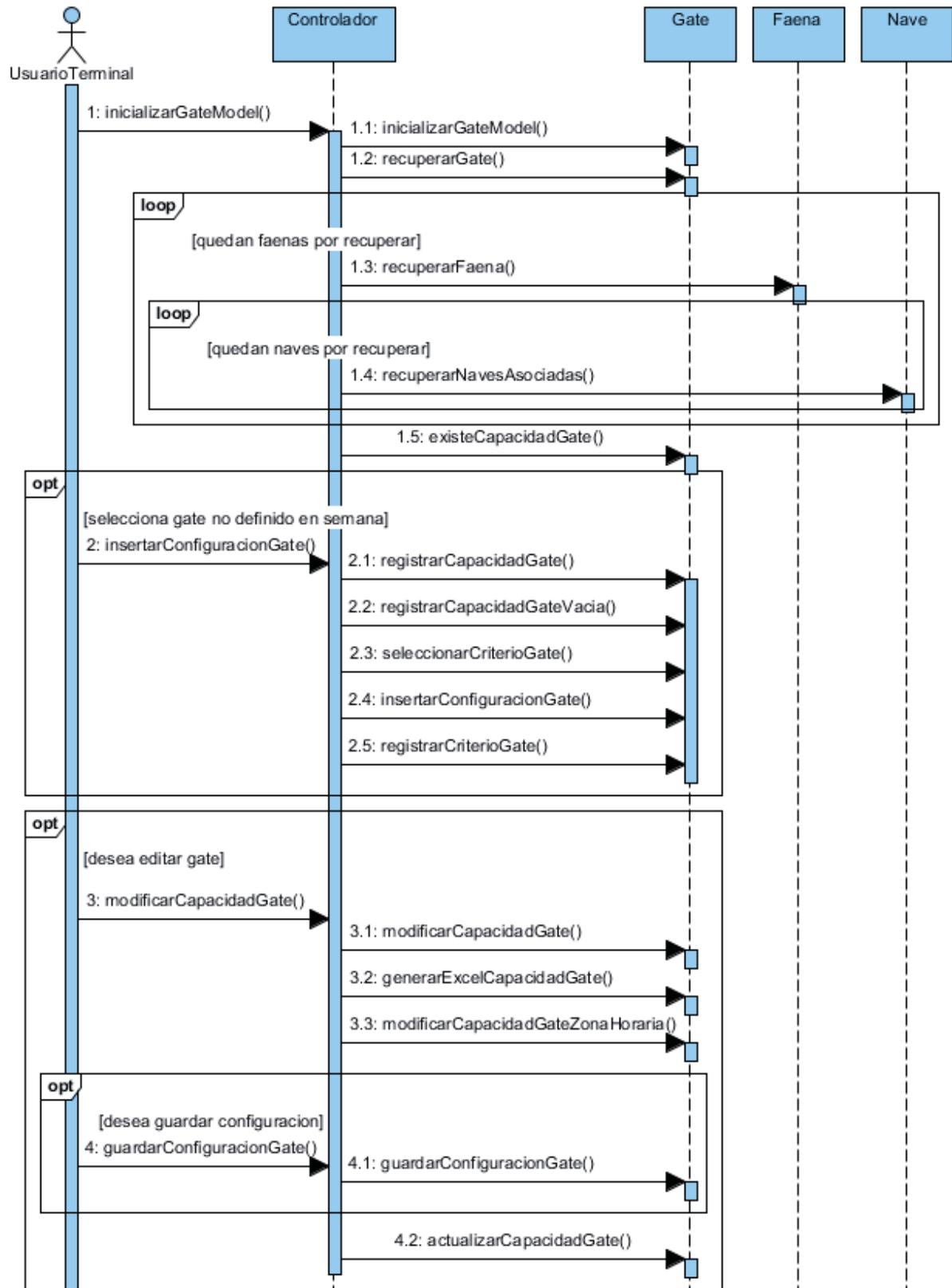


Figura 4.24 – Diagrama de Secuencia Administrar Capacidad de Gate.

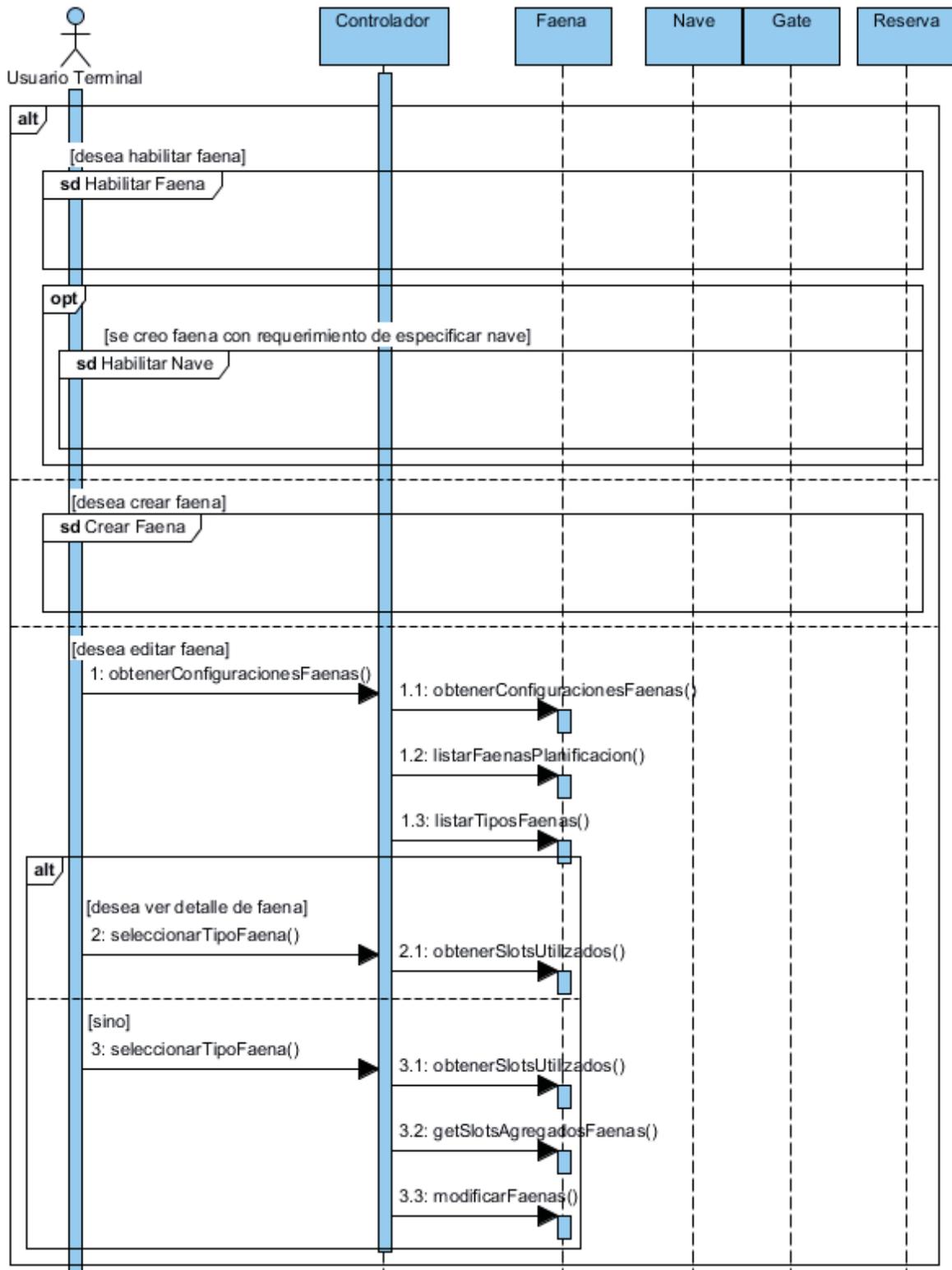


Figura 4.25 – Diagrama de Secuencia Administrar Faenas.

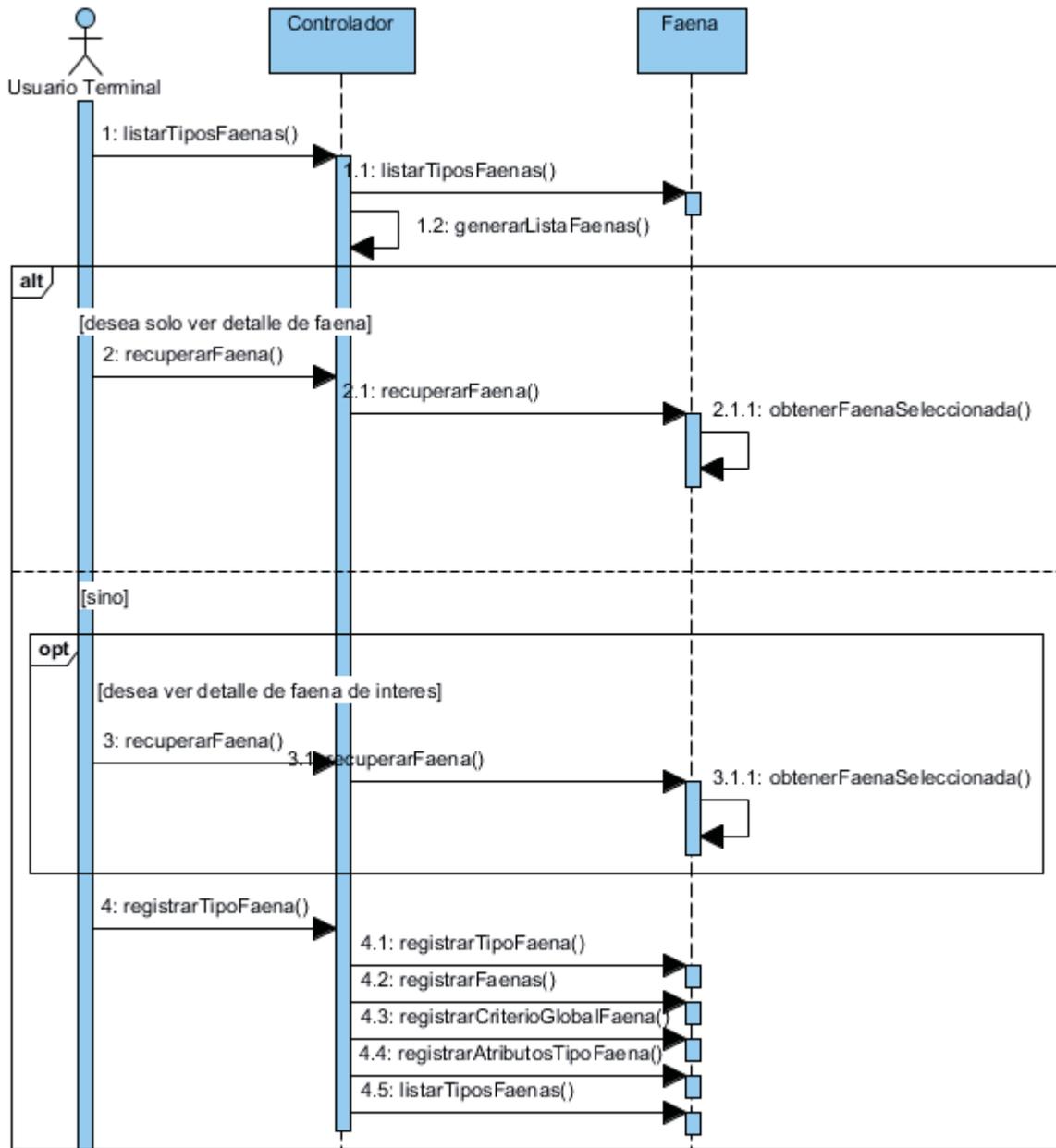


Figura 4.26 – Diagrama de Secuencia Crear Faena.

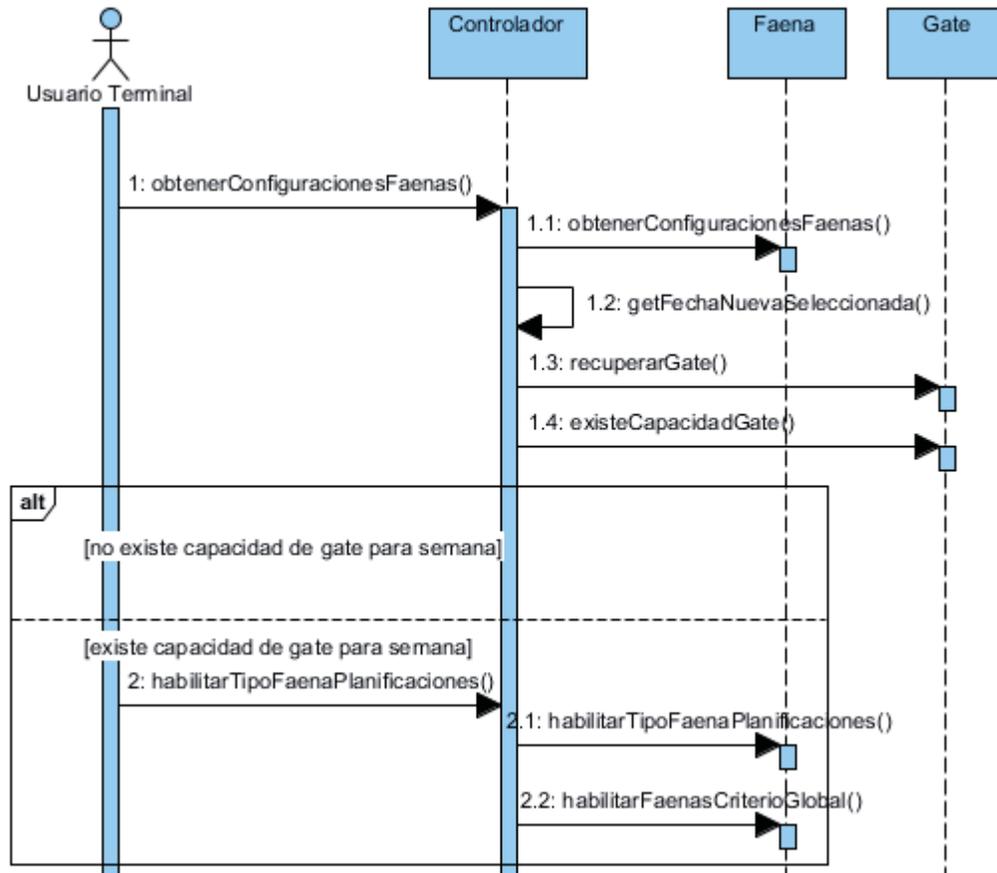


Figura 4.27 – Diagrama de Secuencia Habilitar Faena.

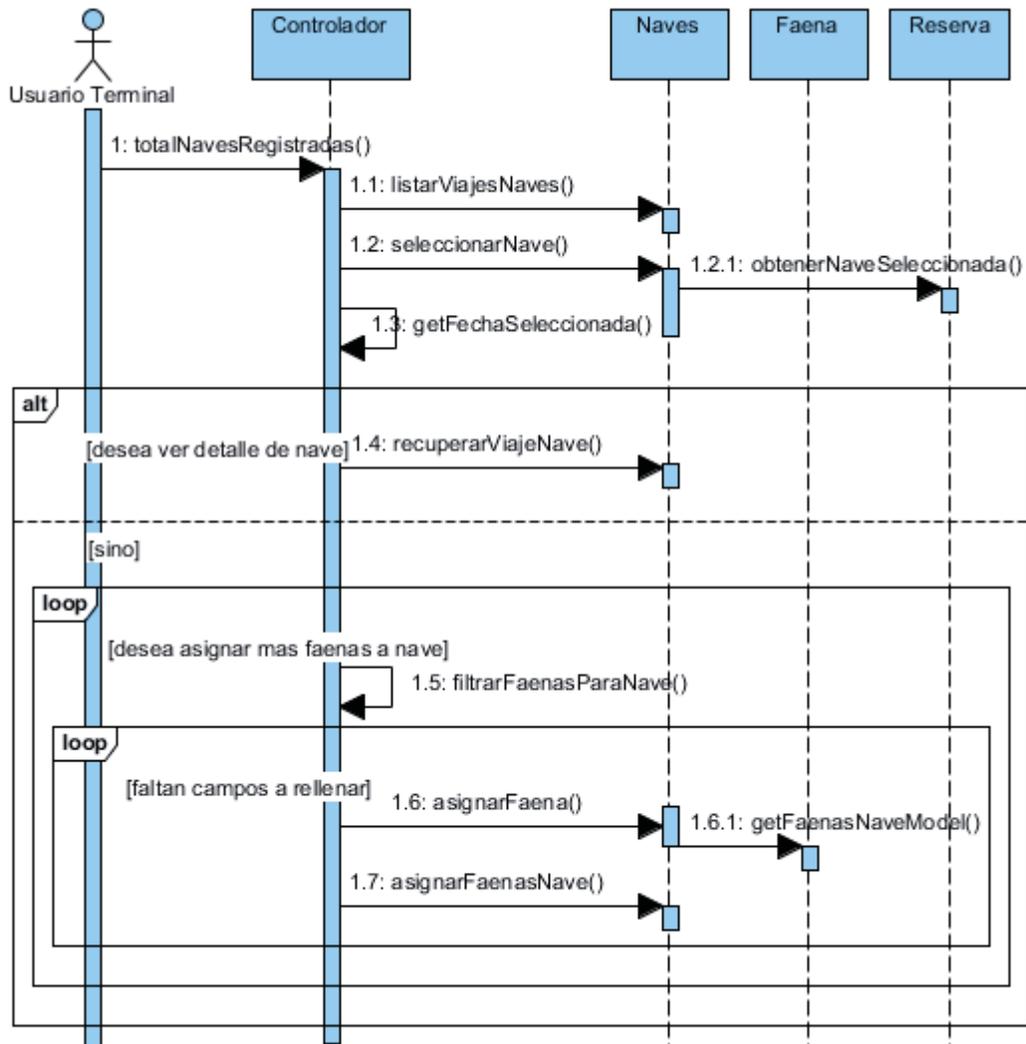


Figura 4.28 – Diagrama de Secuencia Habilitar Nave.

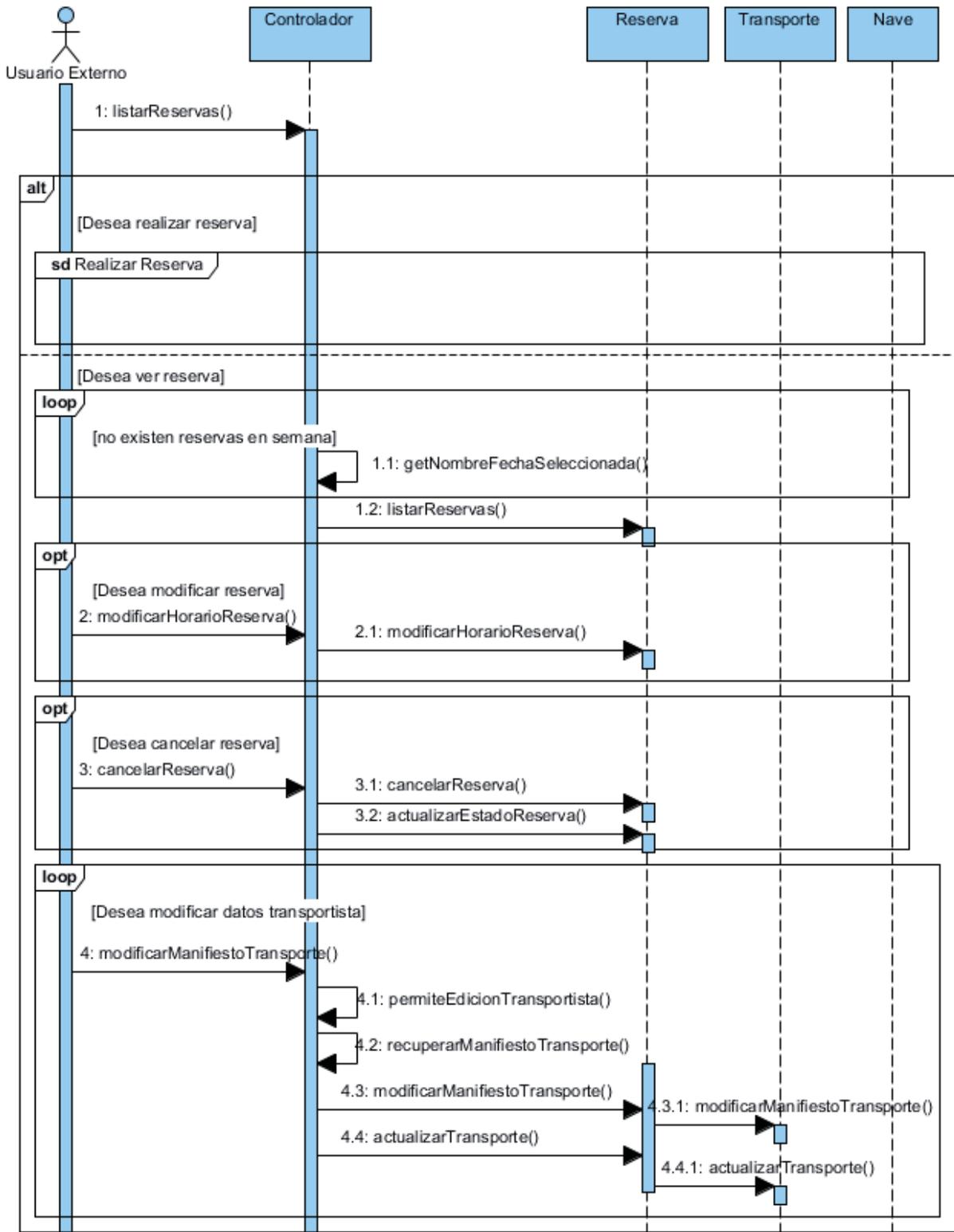


Figura 4.29 – Diagrama de Secuencia Administrar Reserva.

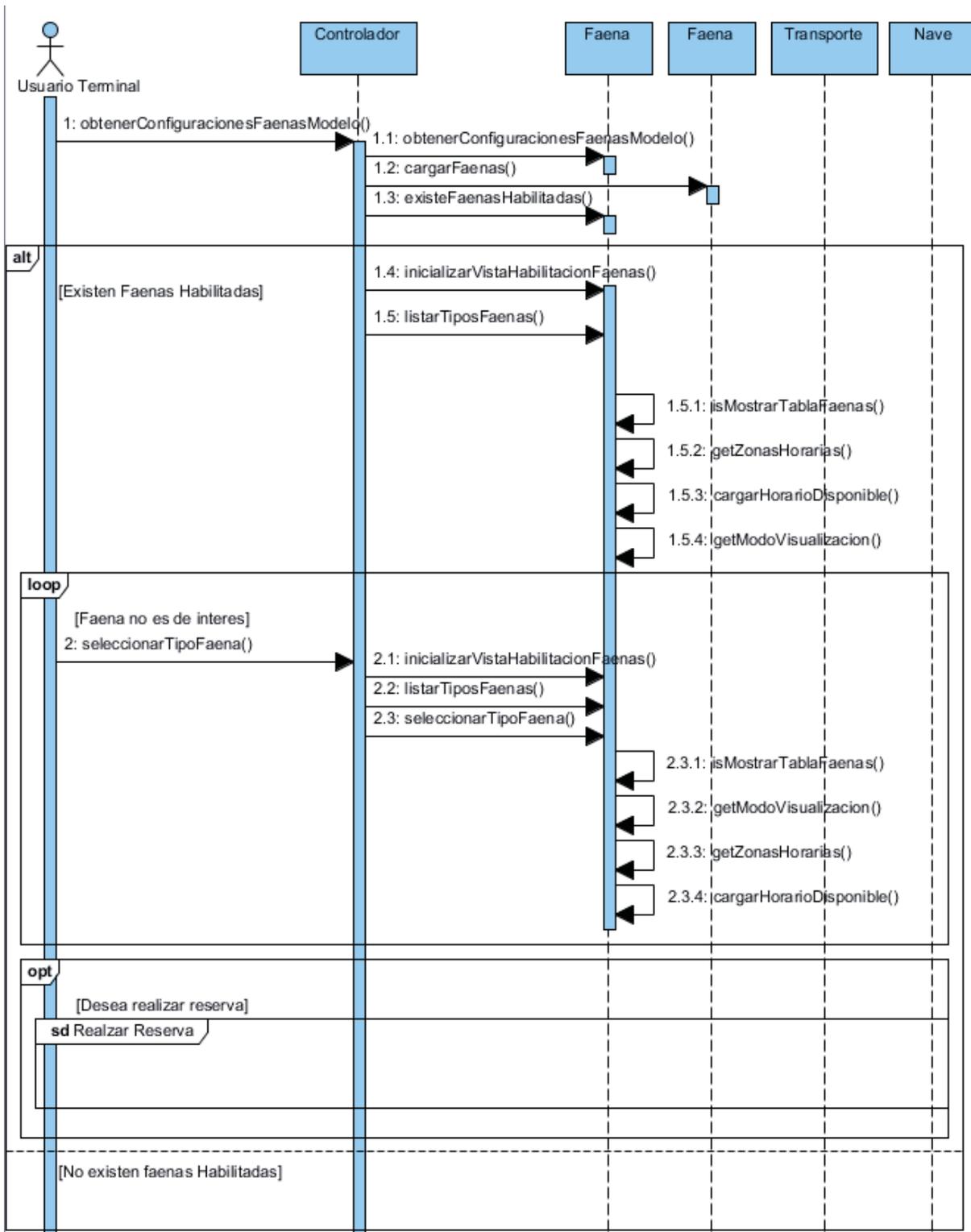


Figura 4.30 – Diagrama de Secuencia Ver Faenas Activas.

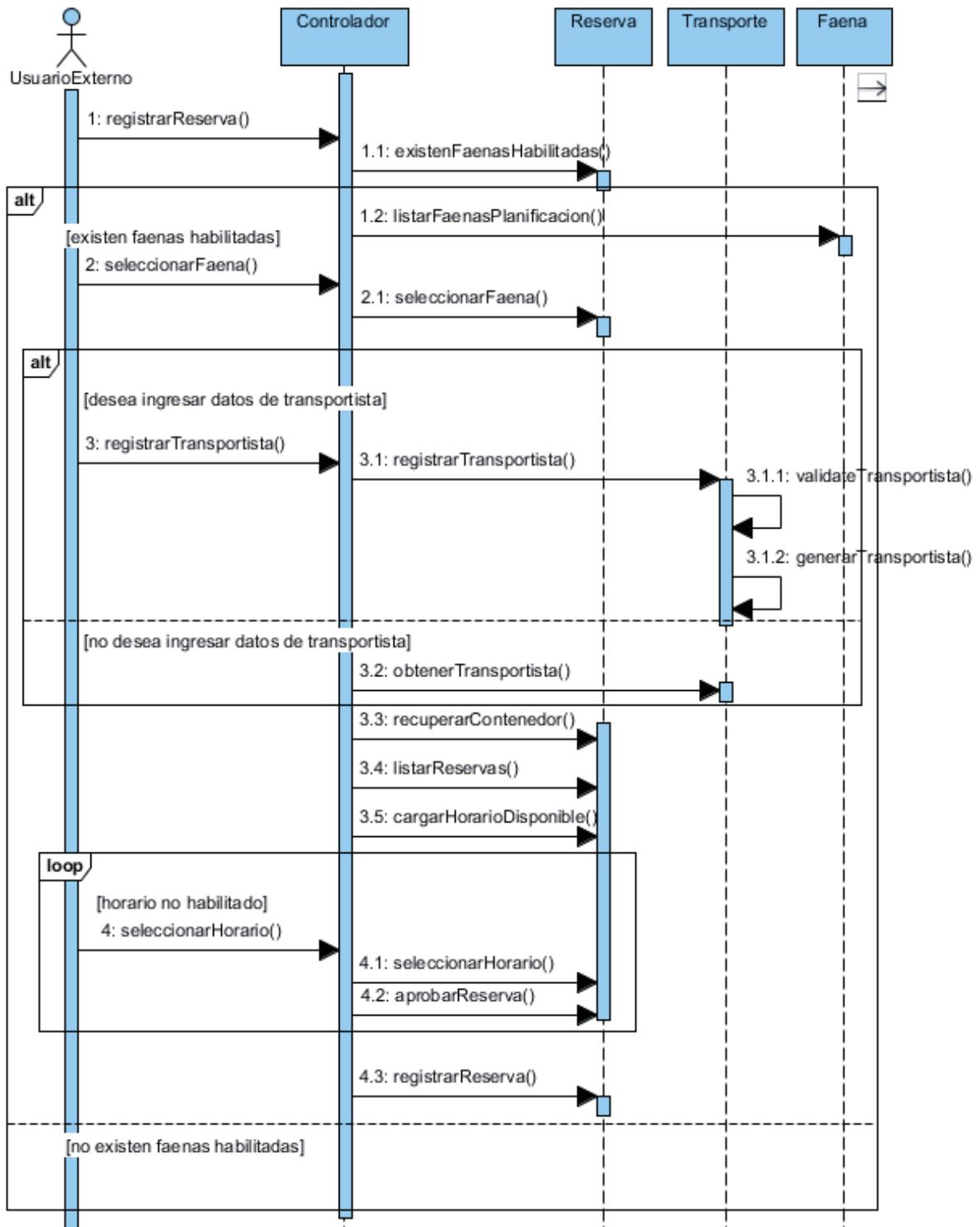


Figura 4.31 – Diagrama de Secuencia Realizar Reservas.

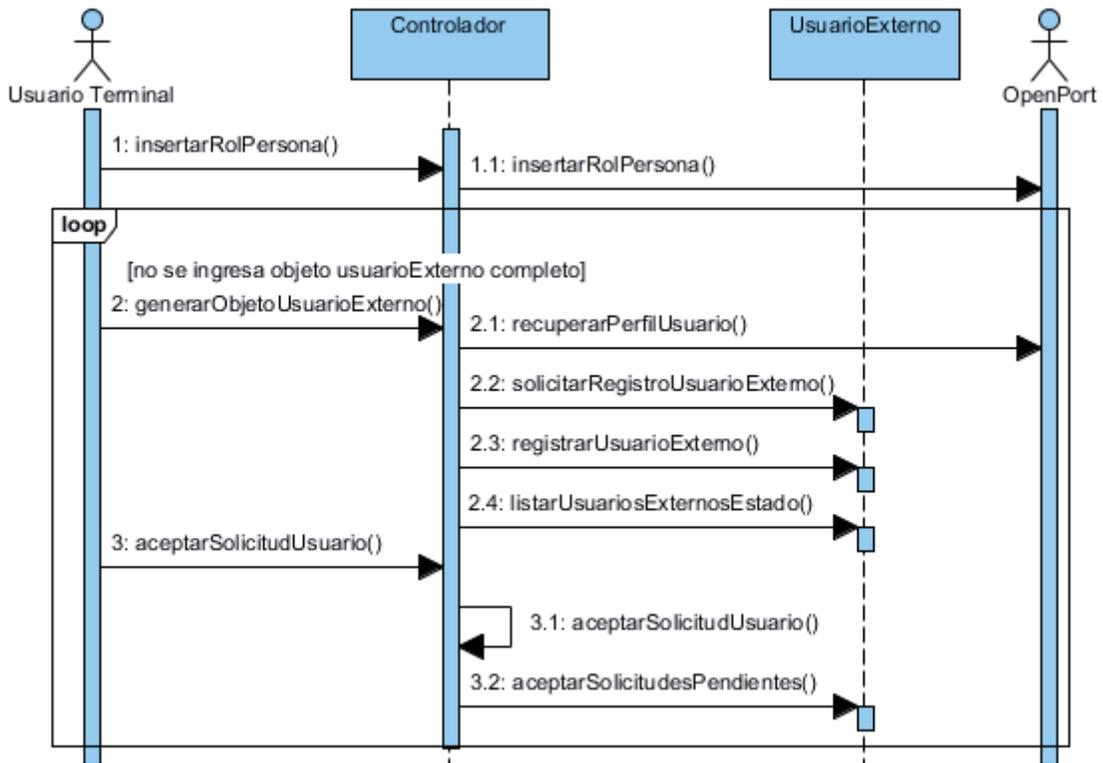


Figura 4.32 – Diagrama de Secuencia Crear Usuario Externo.

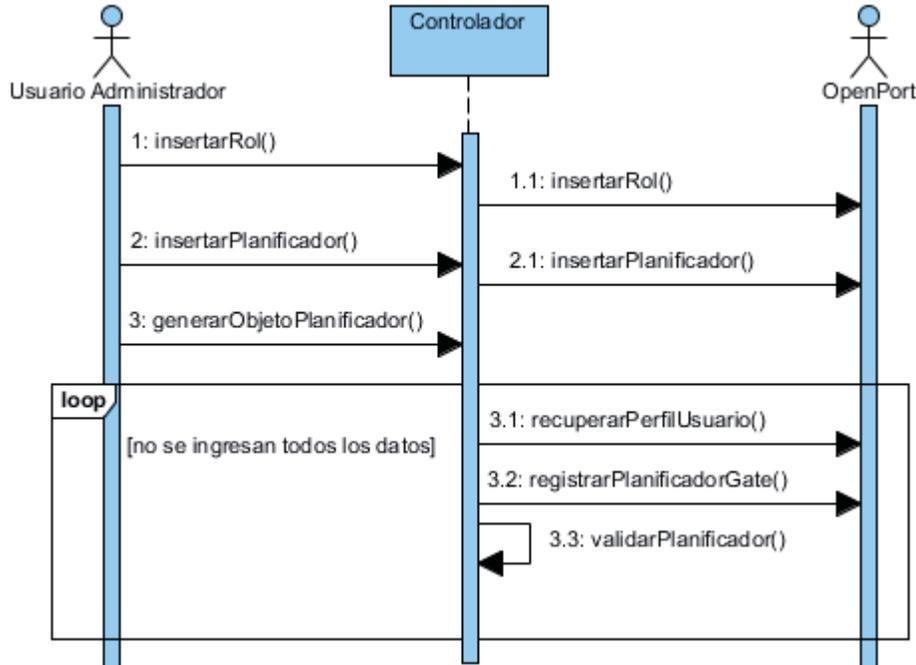


Figura 4.33 – Diagrama de Secuencia Crear Usuario Terminal.

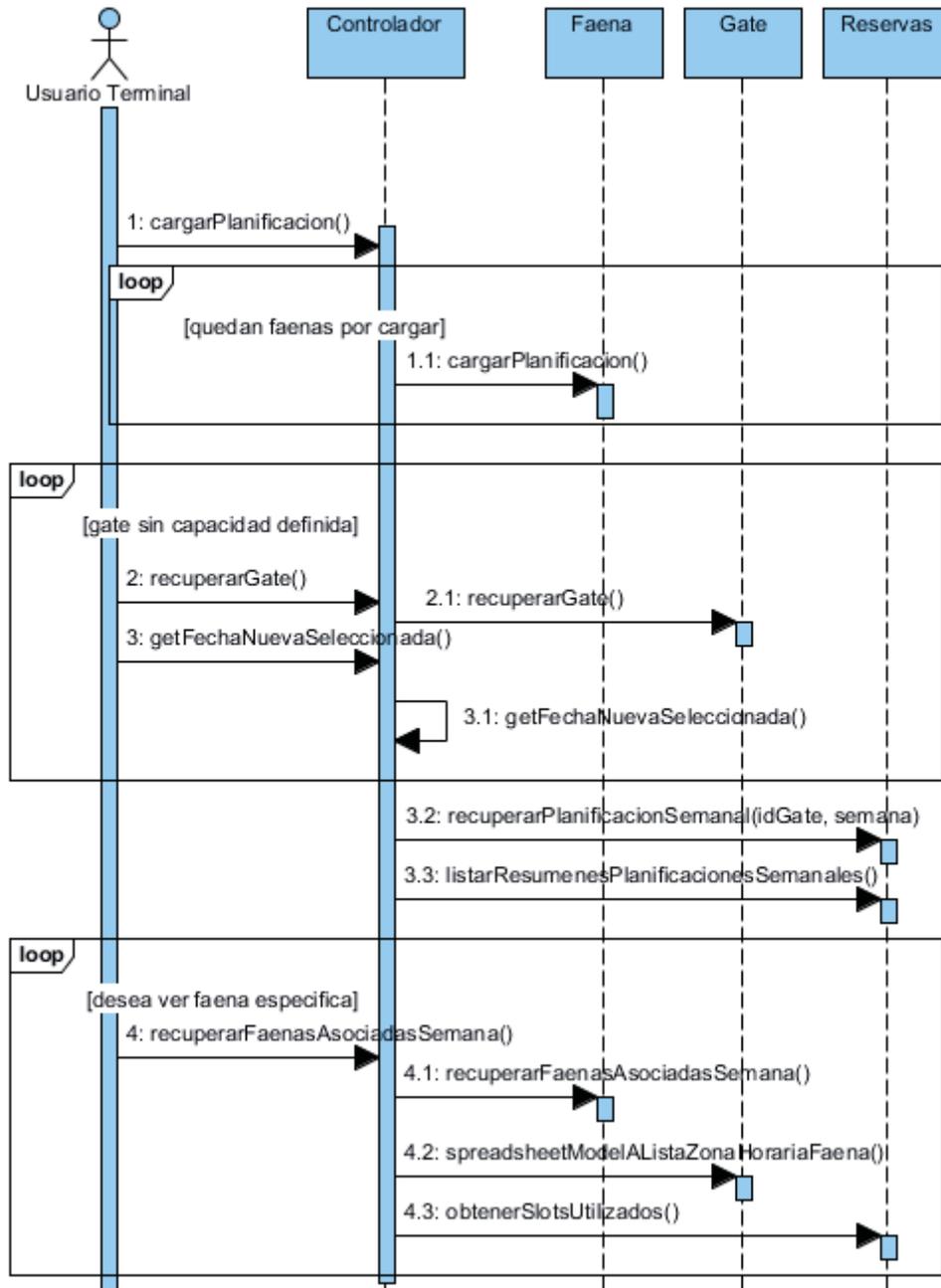
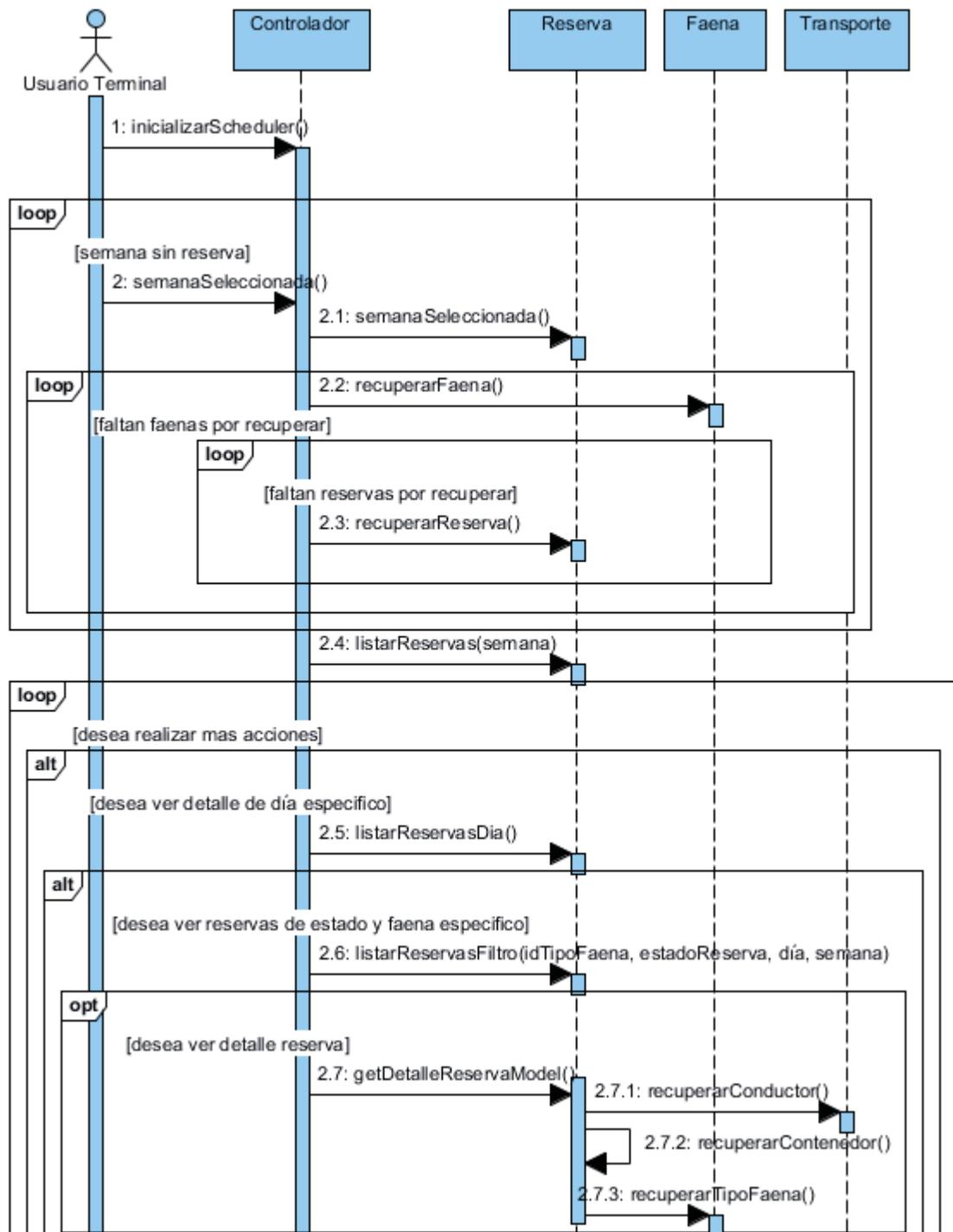


Figura 4.34 – Diagrama de Secuencia Ver Planificación.



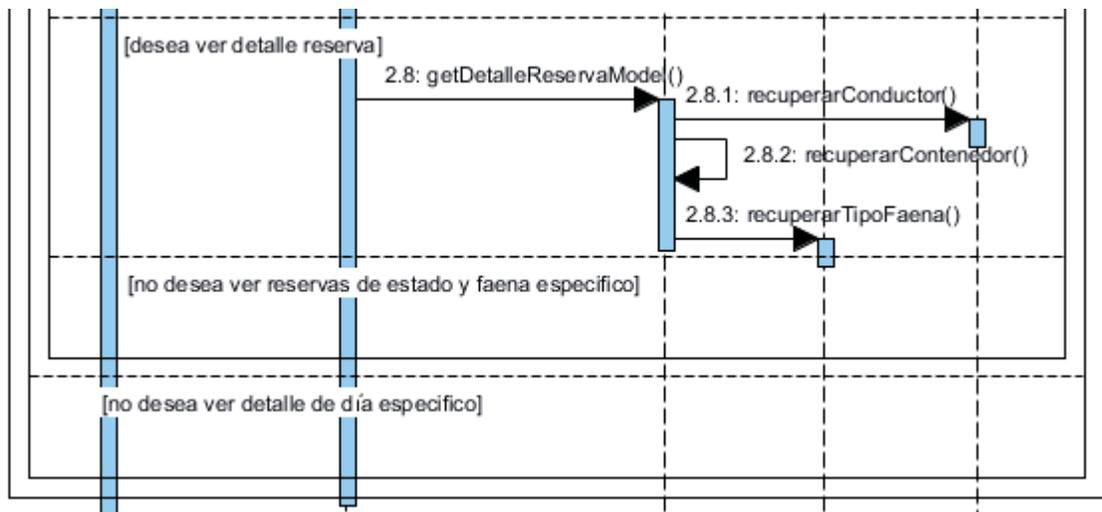


Figura 4.35 – Diagrama de Secuencia Ver Reservas Vigentes.

4.3.4 Diagrama de Interacción Global (DIG)

El Diagrama de Interacción Global es un caso particular de los diagramas de interacciones, este ofrece una visión macro de las interacciones del sistema, organizándolo temporalmente como un Diagrama de Actividades, como cada DSec tiene un marco reutilizable (estos representan los Casos de Uso mencionados en la dimensión funcional), se utilizan de manera de referencias en cada componente.

A continuación se presenta el DIG de OpenGate.

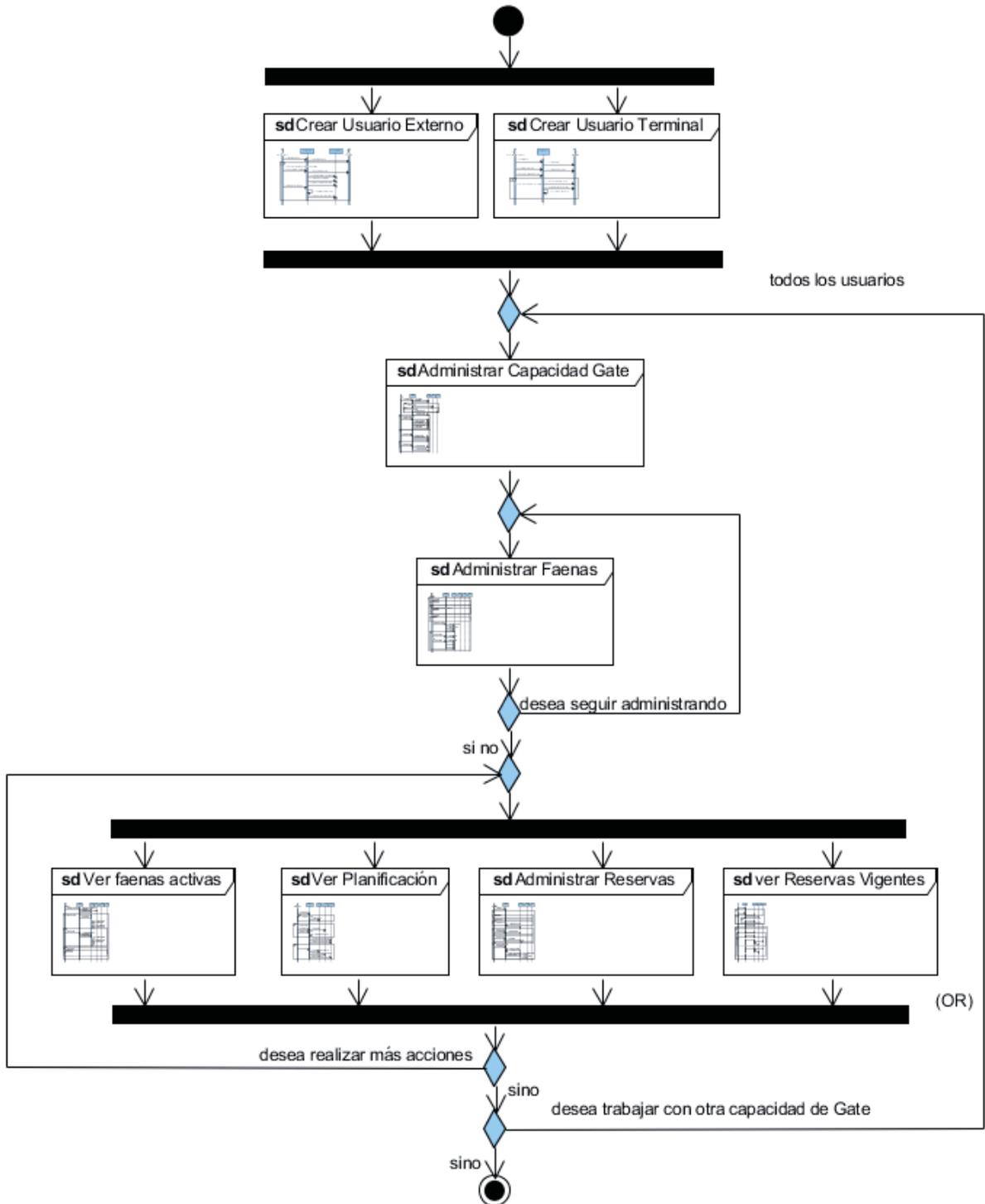


Figura 4.36 – Diagrama de Secuencia Ver Reservas Vigentes.

5 Formulación de propuestas para Proyecto II

Dentro de los objetivos que Proyecto I contempla, el objetivo final consta de proponer posibles directrices para el desarrollo del ramo Proyecto II, ya finalizada la documentación en UML de OpenGate, se pueden establecer con mayor claridad que funcionalidades son deficientes en el modelo, de manera de poder crear proposiciones válidas para la continuidad de este proyecto.

De manera introductoria se muestra la figura 5.1, en donde se ven de manera abstracta el escenario actual de los sistemas y la Figura 5.2 donde se vislumbran las 5 propuestas que se presentan en esta sección.

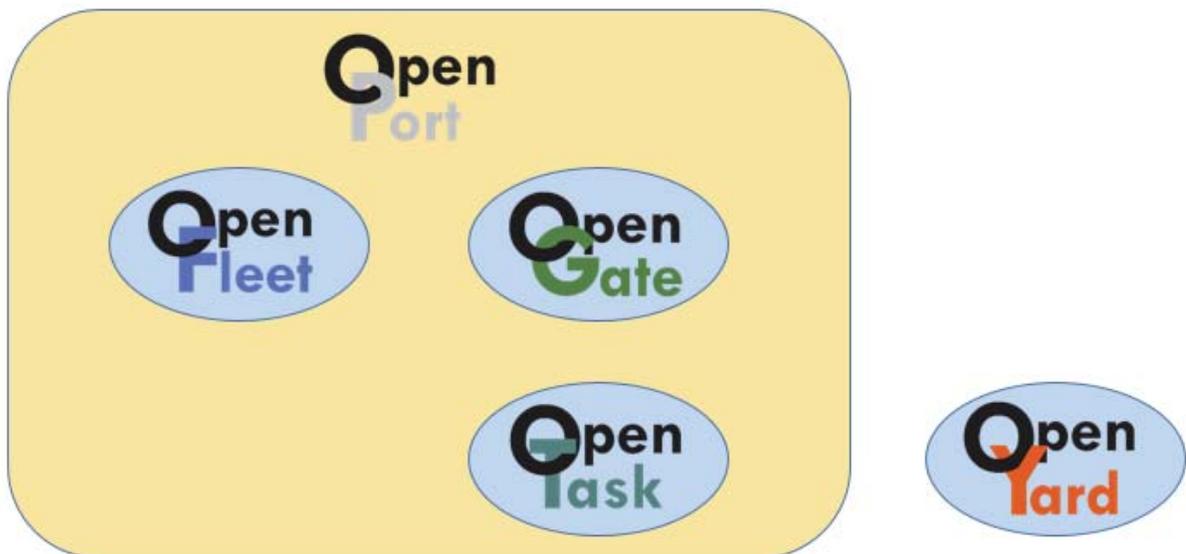


Figura 5.1 – Situación actual Proyecto OpenPort.

En esta figura se aprecia el escenario actual de OpenPort, que si bien fue pensado como un programa de coordinación, investigación y desarrollo de modelos interrelacionados, los primeros prototipos en la etapa inicial del proyecto son los que se lograron concretar, estos son:

- OpenGate (explicado en sección 1, 2, 3 y 4).
- OpenFleet. (explicado en sección 1).
- OpenYard: Herramienta de apoyo en las decisiones asociadas a la gestión de recursos, específicamente de espacio a contenedores para la terminal.
- OpenTask: Herramienta de apoyo a las decisiones de capacidades de recursos del terminal.

Hoy en día, estos sistemas, por sí solos, no abarcan la totalidad de los servicios logísticos, OpenGate solo administra transacciones, OpenFleet coordina la bolsa de oferta y demanda de fletes sin intermediar en pagos, OpenYard anexo al sistema OpenPort solo trabaja internamente. Claramente es necesario avanzar para la conectividad de estos sistemas y generar sinergia, por

lo que se desarrollaron 5 propuestas para seguir con Proyecto que el equipo ve necesarias para poder colaborar y acercarse al principal objetivo de OpenPort, para esto en primera instancia se presenta la Figura 5.2 y luego la Tabla 5.1 donde se especifican brevemente las propuestas.

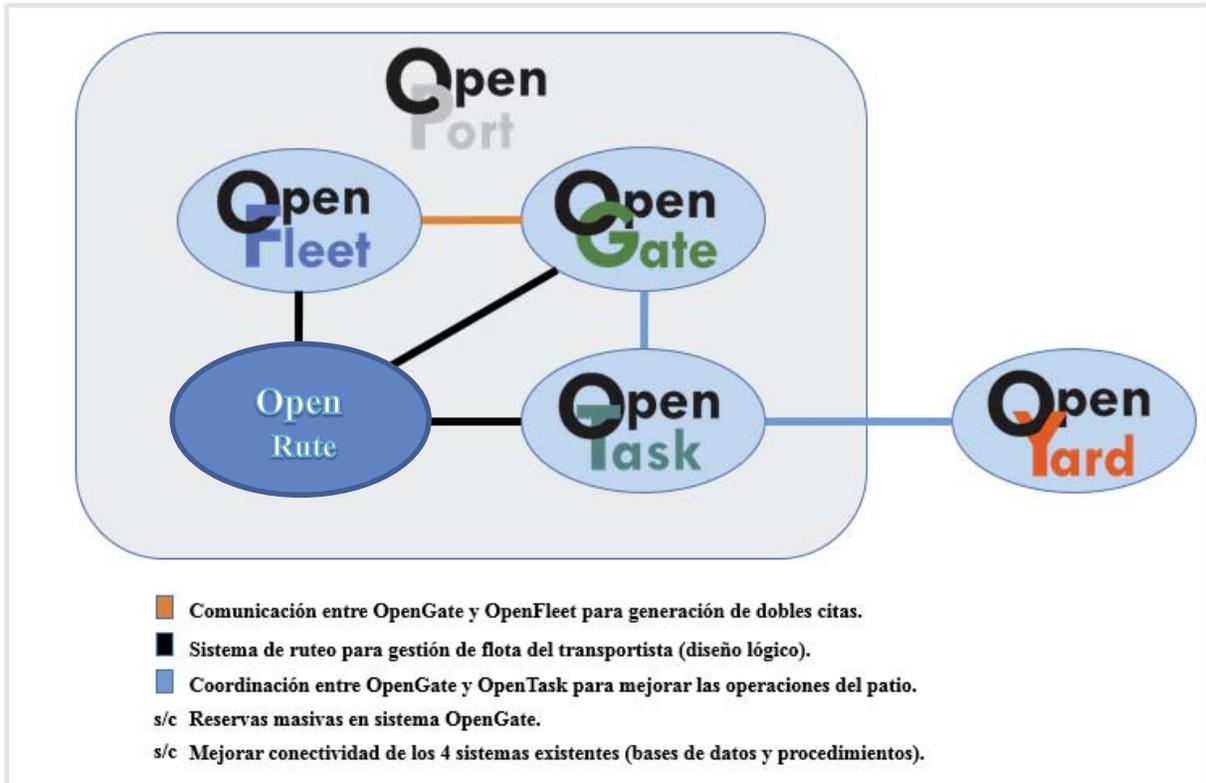


Figura 5.2 – Propuestas de Proyecto II en OpenPort.

Para explicar detalladamente en que consiste cada propuesta explícita en la Figura 5.2 es que se presenta a continuación la Tabla 5.1 donde se expresan las 5 propuestas de pasos a seguir con los tópicos: Propuesta, Resumen e Impactos.

Tabla 5.1 – Tabla resumen de propuestas para Proyecto II.

Propuesta	Resumen	Impactos
Agregar la funcionalidad de dobles citas en el sistema (OpenGate y OpenFleet).	OpenFleet al conectar ofertas y demandas de fletes, puede agregar la función de proponer dobles citas en coordinación con la administración de reservas de OpenGate.	Minimizar viajes muertos por transportista. Mejorar la eficiencia en el patio, mayor certidumbre por llegadas.
Crear diseño lógico de nuevo sistema OpenRut.	Coordinando los 3 sistemas pertenecientes a OpenPort es que se puede generar un DSS cuyo objetivo es la propuesta al transportista para gestionar su flota, como generar dobles citas o como decidir qué hora seleccionar.	Ayuda a la toma de decisiones del transportista.
Generar un heurístico o simulación complementaria a modelo matemático para mejorar la eficiencia de operaciones en el patio (OpenTask).	Dar inteligencia al modelo OpenGate y conectarlo con el modelo OpenTask y OpenYard (en desarrollo), para coordinar la capacidad de Gate y las zonas horarias en función de la oferta de servicios logísticos. Como también administrar esta información coordinada con Opengate para dar bloques horarios continuos en bloques de trabajo de grúas.	Mayor eficiencia para grúas transportadoras de carga contenedorizada. Disminuir incertidumbre sobre llegadas de cargas por bloque.
Agregar funcionalidad de reservas masivas en sistema OpenGate.	OpenGate solo contempla la funcionalidad de reservas por transportista, se omite la posibilidad de generar citas masivas que contengan varios bloques horarios en el prototipo.	Disminuir incertidumbre sobre llegadas masivas. Homogeneizar la oferta de cargas masivas.

Trabajar a nivel global la conectividad de los 4 sistemas.	Definir bases de datos, y procedimientos necesarios para vincular los 4 sistemas en función del principal objetivo de OpenPort, sin desarrollar necesariamente los modelos (propuesta).	Generación de sinergia respecto a los 4 sistemas abordados. Proveer disponibilidad de continuar el proyecto con comunicaciones entre sistemas.
--	---	---

El equipo hoy está decidido por avanzar en la propuesta de mayor interés, esta es “Generar un heurístico o simulación complementaria a modelo matemático para mejorar la eficiencia de operaciones en el patio (OpenTask).”, por lo que en el presente informe se entregarán mayores detalles solo de esta propuesta, a modo de confirmar la primera decisión por parte de los tesistas y en espera de un feedback de los profesores guía y co-guía para determinar un plan de semestre con dicha opción o analizar en mayor detalle alguna otra.

5.1 Detalle de propuesta de mayor interés

Como se mencionó en la primera parte de la sección 5, la propuesta de mayor interés consiste en *Complementar con un heurístico o una simulación el Sistema OpenGate de manera de conectarlo con OpenTask y OpenYard para mejorar la eficiencia de las operaciones en el patio.*

Las hipótesis de esta propuesta se detallan a continuación:

- i. OpenGate se encuentra en el desarrollo de un primer prototipo sólo con las funcionalidades necesarias para administrar las transacciones de citas y reservas.
- ii. OpenGate no contempla las funcionalidades de reporting en ayuda a la toma de decisiones.
- iii. OpenGate no tiene conectividad con ningún otro modelo desarrollado.
- iv. OpenGate deja la administración de la capacidad de Gate solo a la experticia del usuario interno.

De manera continua se presentan las justificaciones de la propuesta:

- i. Dar inteligencia al modelo OpenGate permitirá pronosticar y obtener con mayor precisión la real capacidad de oferta de servicios logísticos del puerto, logrando como objetivo final mejorar la eficiencia de éste.

- ii. Opengate ayuda a predecir llegadas de camiones. Un paso más allá después de la información es ajustar esa demanda a que mejore la operación (OPENTASK), actualmente llegan todos los contenedores por goteo y se ponen en distintos bloques, aumentando las distancias de las grúas, lo que genera mayores costos y pérdidas de tiempo, la proposición del sistema OpenTask es administrar esta información coordinada con OpenGate para dar zonas horarias continuas para contenedores en bloques de trabajo de grúas, disminuyendo la incertidumbre y trabajos extra en bloques del patio.

A continuación se presenta la figura 5.3 que resume los beneficios de un correcto desarrollo de un sistema avanzado de reservas.

Beneficio	Visación Electrónica	Sistema de Reservas	Sistema de Reservas Avanzado
Envío anticipado de información	✓	✓	✓
Menor cantidad de información requerida	✓	✓	✓
Planificación de faenas	✓	✓	✓
Establecimiento de estándares de operación	✓	✓	✓
Control del flujo de ingreso al Puerto		✓	✓
Aumento capacidad ingreso (<u>Gate In</u>)		✓	✓
Mitigación de la congestión		✓	✓
Optimización de la planificación de patio			✓
Mejor comunicación entre actores			✓

Legenda: Beneficio alto: ✓ | Beneficio medio: ✓

24

Figura 5.3 - Beneficios con el uso de sistemas de coordinación y gestión portuaria.

Hoy el primer prototipo de OpenGate se encuentra en la categoría de Sistemas de reservas, en donde impacta de manera alta en 5 de los 9 tópicos analizados, y de manera media en 2 de los 9 tópicos, lo cual no es una evaluación baja como las visaciones electrónicas que se usan en la actualidad, la propuesta propone llegar a un sistema de reservas avanzados donde se impacten de manera alta los 9 tópicos, donde la optimización de la planificación del patio y la mejor comunicación entre actores son metas que se deben cumplir como objetivos finales de la propuesta planteada.

Se analizó la revisión bibliográfica *Benefict of TAS in the servic*, Zehender (2014), en donde mediante programación lineal mixta, experimentos computacionales y simulación con

variables estocásticas con eventos de simulación discreta como aumentar la eficiencia de grúas y camiones en el entorno portuario mejorando el sistema TAS implementado, en donde lo principal consistía en ofrecer pocas capacidades en las ventanas horarias en los tiempos que las grúas se encuentren con mucho trabajo interno, y análogamente ofrecer mayor capacidad en ventanas donde las grúas tengan una mayor disponibilidad, la salida final de este modelo analizado es la determinación de las capacidades y ventanas horarias de sus Gates y la asignación de los camiones y grúas por bloque con el objetivo de reducir tiempos muertos de espera en el terminal.

Si bien esta revisión se aproxima mucho a la propuesta del equipo, existen diferencias con lo planteado anteriormente, la principal diferencia es que OpenGate es un sistema VBS y no TAS, en donde no se podrán asignar los camiones arbitrariamente, si no que se deberá ofrecer zonas horarias al usuario reservador, por lo que el modelado del heurístico o análisis de impactos deberá plantearse de una manera distinta a la encontrada.

6 Conclusiones

En este informe final de proyecto, el equipo culminó la realización de la documentación del prototipo OpenGate, ya obtenida una visión macro del sistema a abordar, se trabajó de manera íntegra en la documentación total de la plataforma tecnológica de reservas on-line OpenGate, llegando a concluir la totalidad de los Diagramas (6 totales) para el fin del proyecto, estos son: DCla, DCU DME DIG DSec y Dact y la documentación complementaria DoCU.

La revisión literaria permitió conocer cómo se abordan las problemáticas más comunes en los puertos del mundo, como también tener información de soluciones parecidas a la plataforma a documentar, por ejemplo, Australia (one-stop), Estados Unidos (L.A Long Beach, e-modal), trabajan con plataformas similares a OpenGate, como se aprecia en el benchmark de la sección 2.3, los efectos positivos de implementación de un sistema VBS en el ambiente portuario, lo que corrobora una necesidad de mayor control en las terminales portuarias e integración de los distintos agentes actuantes para disminuir la incertidumbre y aumentar la eficiencia del puerto de San Antonio.

En cuanto al trabajo en el semestre, se realizó una carta Gantt con la finalidad de orientar el trabajo y proponer metas de entrega para la correcta realización del proyecto en su completitud, esta fue respetada y se aprovecharon las holguras que se establecieron, específicamente la brecha que se permitió para aplicar criterios de calidad e integración de los modelos. Los objetivos se cumplieron, ya que se obtuvo los modelos UML documentados e integrados del sistema, esto gracias a la coordinación y reunión con el equipo de informáticos que desarrollaron el prototipo virtual, lo que permitió evitar ambigüedades en el entendimiento de las funcionalidades de OpenGate.

Finalmente, la decisión de documentar un sistema descriptivo y conceptual ayudó a entender en primera instancia, la esencia del sistema OpenGate, esto incluye conocer profundamente que es lo que hace el sistema, y en segunda instancia poder desarrollar mejoras para el trabajo de Proyecto II.

Bibliografía

- Ascencio, Luis (2012). **Formulación General – Línea 2 CORFO: Proyecto I+D Aplicada**. Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso.
- Escuela de Ingeniería Industrial (2014). **Informe de presentación de informes**. Abr. Disponible vía Web en <http://www.eii.ucv.cl>
- San Antonio Terminal Internacional (2014). **Operaciones**. Abr. Disponible vía Web en <http://www.stiport.com>.
- Empresa Portuaria San Antonio (2014). **Sistema Portuario, Estadísticas**. Abr. Disponible vía Web en <http://www.epsa.cl>.
- CEPAL (2010). **Boletín FAL Edición N° 282**. Ene. Disponible vía Web en <http://www.cepal.org/usi/noticias/botfall/5/41565/FAL-282-WEB.pdf>
- CEPAL (2012). **Boletín FAL Edición N° 314, número 10 de 2012**. May. Disponible vía Web en <http://www.eclac.org/Transporte/noticias/bolfall/3/48953/FAL-314-WEB.pdf>
- Revista myt (2013) **Un débil eslabón, Informe sobre puertos de la región**. May. Disponible vía Web en <http://revistamyt.com/wp-content/uploads/2013/04/PUERTOS.pdf>
- Bustos, Guillermo (2010). **Modelos para Sistemas de Información**. Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso.
- Corleone (2013). **Prototipo OpenGate**. Disponible vía Web en <http://www.corleone.cl/opengate/>
- Zehender (2014). **Benefict of TAS in the servic**.
- Chen (2013). **Managing truck arrivals with time windows container terminal**.
- Mwemezi (2012). **Inland container depot integration into logistic networks**.
- Van Aspere (2011). **Evaluating impact of truck announcements on container stacking efficiency**.
- Zhang (2010). **Heuristic-based truck scheduling for inland container transportation**.
- Huynh (2011). **Assesing Truck delays using webcams**.
- Franz (2012). **Performance Analysis of Slot-based Appointment at Air Cargo Terminal**

Anexo 1 – Clase Reserva

a	Reserva
-gate : Gate	
-id : Integer	
-faenas : List<Faena> = new ArrayList<Faena>()	
-serialVersionUID3 : long = -6690503297605413366L	
-reserva : Reserva	
-informacionTransportista : Boolean = false	
-informacionGeneral : Boolean = false	
-informacionCarga : Boolean = false	
-informacionNave : Boolean = false	
-capacidadGate2 : int	
-reservas : List<Reserva>	
-serialVersionUID2 : long = -401948314656380054L	
-slotsHabilitados : int	
-serialVersionUID : long = 304562263351399445L	
-capacidadGate : CapacidadGate	
-documentoAduanero : String	
-fechaCreacion : Date	
-estadoReserva : EstadoReserva	
-codigo : String	
-id2 : Integer	
-contenedor : Contenedor	
-documentoNaviero : String	
-tipoCarga : TipoCarga	
-idRepresentante : Integer	
-idReservador : Integer	
-peso : Float	
-faena : Faena	
-numeroBultos : Integer	
-tipoBulto : TipoBulto	
-estadoTransporte : boolean	
-manifiestoTransporte : ManifiestoTransporte	
-dia : Integer	
-zonaHoraria : Integer	
-tipoCamion : TipoCamion	
-estadoVisacion : boolean	
-operadorNaviero : String	
-viajeNave : ViajeNave	
-serialVersionUID4 : long = -4175447525867764132L	
-estadoCamion : EstadoCamion	
-semana : Date	
-planificacionSemanal : PlanificacionSemanal	
-zonaHorariaSeleccionada : int	
-gates : List<Gate>	
-reservasDiaSeleccionada : ReservasDia	
-reservasDia : List<ReservasDia>	
-reservasSemanaGate : List<ReservasZonaHoraria>	
-diaSeleccionado : int	
-reservaSeleccionada : Reserva	
-reservasZonaHorariaSeleccionadas : List<Reserva>	
-idGateSel : Integer	

```

-reservasSemana : List<Reserva>
<<Constant>> -EJECUTADA =
"Ejecutada"
-friendlyName : String
<<Constant>> -PENDIENTE = "Pendiente"
<<Constant>> -APROBADA = "Válida"
<<Constant>> -PENDIENTE_INCOMPLETA = "Pendiente e Incompleta"
<<Constant>> -INCOMPLETA =
"Incompleta"
<<Constant>> -CANCELADA = "Cancelada"
-zonaHoraria2 : int
-seriaVersionUID6 : long = 2247435756622110188L
-reservasDia2 : Map<Integer, ReservasDiaZonaHoraria>
-semanaSeleccionada : Date
-seriaVersionUID5 : long = 2839308443162976817L
-scheduleModel : DefaultScheduleModel = new DefaultScheduleModel()
-reservaSeleccionada2 : boolean = false
-scheduleEvent : ScheduleEvent = new DefaultScheduleEvent()
-diaSeleccionado2 : Integer
-estadoReservaSeleccionada : EstadoReserva = null
-reservaSeleccionada3 : Reserva
-reservas2 : List<Reserva>
-dia3 : Date
-faenasHabilitadas : int
-reservasEjecutadas : int
-reservasCanceladas : int
-semana3 : Date
-diaNombre : String
-seriaVersionUID10 : long = -985310887100979614L
-idFaenaSeleccionada2 : Integer = null
-tiposFaenas : List<TipoFaena>
-HORAS_PREMIUM : int = 8
-EVENTO_GRIS : String = "evento-gris"
-codigo2 : String
-seriaVersionUID11 : long = 8618001253202605830L
-slotsDisponiblesZona : List<SlotsDisponiblesZona> = new ArrayList<SlotsDisponiblesZona>()
-zonaHorariaContador : int = 0
-timeZone : TimeZone
-seriaVersionUID7 : long = 82088436341838790L
-LOGGER : Logger = LoggerFactory
.getLogger(ScheduleReservaModel.class)
-fechaFinalAnterior : Date
-fechaInicialAnterior : Date
-navePuerto : NavePuerto
-EVENTO_CAFE : String = "evento-cafe"
-EVENTO_AMARILLO : String = "evento-amarillo"
-EVENTO_ROJO : String = "evento-rojo"
-semana2 : Date
-zonasHorarias : int
-scheduleReservaHorarioModel : ScheduleReservaModel = new ScheduleReservaModel()
-dia2 : Integer

```

```

-reserva2 : Reserva
-LOGGER2 : Logger = LoggerFactory
.getLogger(ReservaHorarioModel.class)
-reservasSemana2 : List<Reserva>
-slotsDisponiblesZona2 : List<SlotsDisponiblesZona> = new ArrayList<SlotsDisponiblesZona>()
-zonaHoraria3 : Integer
-seriaVersionUID8 : long = -2358905872595507047L
-minutosPorSlot : int
-zonaHorariaSeleccionada2 : Integer
-semanaSeleccionada2 : Date
-idRepresentante2 : Integer
-atributoDocumentoAduanero : Boolean = false
-reservaRepresentante : boolean = false
-usuarioRepresentador : boolean = false
-atributoViajeNave : Boolean = false
-faenaSeleccionada : Faena
-idFaenaSeleccionada : Integer
-operadorNaviero2 : String
-contenedor2 : Contenedor = new Contenedor()
-documentoNaviero2 : String
-faenasSemana : List<Faena>
-idCamionSeleccionado : Integer
-atributoOperadorNaviero : Boolean = false
-camiones : List<Camion>
-transportistaSeleccionado : Transportista
-idTipoBultoSeleccionado : Integer
-atributoDatosCargas : Boolean = false
-atributoContenedor : Boolean = false
-idRepresentado : Integer
-usuariosExternos : List<UsuarioExterno>
-documentoAduanero2 : String
-viajesNaves : List<ViajeNave>
-idViajeSeleccionado : Integer
-idConductorSeleccionado : Integer
-seriaVersionUID12 : long = 1606625786378588157L
-slots : int[]
-zonaHoraria4 : Integer
-conductores : List<Conductor>
-camionSeleccionado : Camion
-patenteCamionSeleccionado : String
-atributoTipoBulto : Boolean = false
-scheduleReservaModel : ScheduleReservaModel = new ScheduleReservaModel()
-idReserva : Integer
-requisitosVisibilidad : boolean = true
-diaSeleccionado3 : Integer
-reservasSemana3 : List<Reserva>
-contenedorValidado : boolean = false
-modalidadSeleccionada : ModalidadTipoFaena
-idGateSel2 : Integer
-gates2 : List<Gate>
-reservas3 : List<Reserva>
-seriaVersionUID13 : long = -8874261364841978969L
-semana4 : Date
-idGateSel3 : Integer

```

```

-gates3 : List<Gate>
-semanaSeleccionada3 : Date
-scheduleReservasModel : ScheduleReservasModel
-idTransportistaSeleccionado : Integer
-transportistas : List<Transportista>
-atributoDocumentoNaviero : Boolean = false
-atributoTipoCamion : Boolean = false
-atributoTipoCarga : Boolean = false
-atributoBulto : Boolean = false
-estadoCamionSeleccionado : EstadoCamion
-tipoCamionSeleccionado : TipoCamion
-tipoCargaSeleccionado : TipoCarga
-numeroBultos2 : Integer
-tiposBultos : List<TipoBulto>
-peso2 : Float
-conductorSeleccionado : Conductor
-incluirInformacionTransportista : Boolean = false
-atributoCantidad : Boolean = false
-atributoPeso : Boolean = false
-seriaVersionUID9 : long = -7858981430820347575L
-horarioEditable : Boolean = false
-LOGGER3 : Logger = LoggerFactory
.getLogger(ScheduleReservasModel.class)
-seriaVersionUID14 : long = 1892259173997609472L
-reservaSeleccionada4 : Reserva
-model : DefaultScheduleModel
-minutosPorSlot2 : int
-transportistaEditable : Boolean = false
-atributoNave : Boolean = false
-zonaHorariaSeleccionada3 : Integer

+ReservasDiaZonaHoraria(reservas : List<Reserva>, slotsHabilitados : int, capacidadGate : int)
+getEstadosCamiones() : EstadoCamion []
+setRequisitosVisibilidad(requisitosVisibilidad : Boolean) : void
+obtenerNavePuertoSeleccionado() : NavePuerto
+obtenerTipoBultoSeleccionado() : TipoBulto
+getRequisitosVisibilidad() : Boolean
+isUsuarioRepresentador() : boolean
+completarCamiones(patente : String) : List<String>
+obtenerConductor() : Conductor
+obtenerCamion() : Camion
+obtenerNaveSeleccionada() : ViajeNave
+isReservaRepresentante() : boolean
+getTiposCamiones() : TipoCamion []
+seleccionarCamion() : void
+obtenerTransportista() : Transportista
+isContenedorValidado() : boolean
+ReservasDiaZonaHoraria()

```

```

+isReservaCancelada() : boolean
+isEstadoVisacion() : boolean
+reservasSemanaCargadas() : boolean
+requisitosCumplidosVerReservas() : boolean
+reservasDiaCargadas() : boolean
+reservasSemanaGateCargadas() : boolean
+getReservasDiaZonaHoraria(dia : int) : ReservasDiaZonaHoraria
+seleccionarReservaDia(reserva : ReservasDia) : void
+isEstadoTransporte() : boolean
+existeRepresentante() : boolean
+friendlyName() : String
-EstadoReserva(friendlyName : String)
-disponibilidadReservar(fecha : Date) : boolean
+onDateSelect(selectEvent : SelectEvent) : void
+crearEventoReservaOriginal(dia : Integer, zonaHoraria : Integer) : void
+getFechaSeleccionada() : String
+onEventMove(event : ScheduleEntryMoveEvent) : void
+addEvent() : void
+onEventResize(event : ScheduleEntryResizeEvent) : void
+getZonaHoraria() : Integer
+setDia(dia : Integer) : void
+isReservaSeleccionada() : boolean
+crearEventoNoDisponible(dia : int, zonaHoraria : int, timeZone : TimeZone) : void
+setZonaHoraria(zonaHoraria : Integer) : void
-seleccionarDiaZona(fechaSeleccionada : Date) : void
+getDia() : Integer
+obtenerNavePuerto() : NavePuerto
+getFechaNuevaSeleccionada(timeZone : TimeZone) : String
+getFechaActualSeleccionada(timeZone : TimeZone) : String
+getScheduleModel() : ScheduleModel
+obtenerEstadosReserva() : EstadoReserva []
+obtenerFechaReservada(reserva : Reserva, timeZone : TimeZone) : String
+onEventSelect(event : SelectEvent) : void
+ScheduleReservasModel(events : List<ScheduleEvent>, minutosPorSlot : int)
+ScheduleReservasModel()

```

Anexo 2 – Clase Usuario Externo

UsuarioExterno
<pre> <<Constant>> -AGENTE_ADUANA = "Agente de Aduana" -friendlyName : String <<Constant>> -REPRESENTANTE_TRANSPORTE_INTERNACIONAL = "Representante del Transporte Internacional" <<Constant>> -FREIGHT_FORWARDER = "Freight Forwarder" -usuarioExternoSeleccionado : UsuarioExterno -usuariosExternos : List<UsuarioExterno> -seriaVersionUID2 : long = -7593258730156258751L <<Constant>> -ALMACENISTA = "Almacenista" <<Constant>> -IMPORTADOR = "Importador" -telefono : String -giroComercial : GiroComercial -correo : String -direccion : String -codigoPostal : String -zonasHorarias : List<String> -password : String -rut2 : String -nombreEmpresa2 : String -LOGGER : Logger = LoggerFactory .getLogger(UsuarioExternoModel.class) -codigoPostal2 : String -direccion2 : String -requisitoMapa : boolean = false -zonaHoraria : String -usuario : String -giroComercial2 : GiroComercial -telefono2 : String -correo2 : String -seriaVersionUID3 : long = 2592867436615925540L -coordenadaUbicacion : Coordenada -ciudades : List<Lugar> -ciudadSeleccionada : Lugar -idCiudadSeleccionado : Integer -perfilUsuario : PerfilUsuario -id : Integer -seriaVersionUID : long = 2216901908612158106L -rut : String -ciudad : Lugar -nombreEmpresa : String <<Constant>> -EXPORTADOR = "Exportador" </pre>
<pre> +friendlyName() : String +seleccionarUsuario(usuarioExterno : UsuarioExterno) : void -GiroComercial(friendlyName : String) +getGirosComerciales() : GiroComercial [] +autocompletarCiudad(solicitud : String) : List<Lugar> +actualizarRequisitoMapa() : void +seleccionarCiudad() : void +isRequisitoMapa() : boolean </pre>

Anexo 3 – Clase Transporte

Transporte
-patente : String
-camion : Camion
-conductor : Conductor
-transportista : Transportista
-idReserva : Integer
-transportistaSeleccionado : Transportista
-transportistas : List<Transportista>
-serialVersionUID3 : long = 2841692063883637511L
-idReserva2 : Integer
-horarioEditable2 : Boolean = false
-transportistaEditable2 : Boolean = false
-camionSeleccionado : Camion
-idCamionSeleccionado : Integer
-camiones : List<Camion>
-transportistas2 : List<Transportista>
-conductorSeleccionado : Conductor
-incluirInformacionTransportista : Boolean = false
-rut2 : String
-nombre2 : String
-serialVersionUID6 : long = -3838429820739388982L
-idTransportista2 : Integer
-id4 : Integer
-serialVersionUID5 : long = -2481864466100517032L
-conductores2 : List<Conductor>
-conductorSeleccionado2 : Conductor
-serialVersionUID7 : long = -8735807130746608284L
-idConductorSeleccionado : Integer
-conductores : List<Conductor>
-transportistaSeleccionado2 : Transportista
-idTransportistaSeleccionado : Integer
-id2 : Integer
-serialVersionUID2 : long = 1616670215273944719L
-id3 : Integer
-serialVersionUID4 : long = -3564063693873595666L
-rut : String
-nombre : String
-horarioEditable : Boolean
-transportistaEditable : Boolean
-idTransportista : Integer
-id : Integer
-serialVersionUID : long = 5202819254946624611L
+obtenerConductor() : Conductor
+obtenerCamion() : Camion
+obtenerTransportista() : Transportista

Anexo 4 – Clase Gate

Gate
-ultimaModificacion : Date
-serialVersionUID : long = 446062387064958496L
-planificadorGate : PlanificadorGate
-totalNavesAsociadas : int
-todosGatesHabilitados : boolean
-ultimaModificacion2 : Date
-capacidadGateDefinida : boolean
-serialVersionUID4 : long = 2501948869695819997L
-semana2 : Date
-faenasExpoHabilitadas : int
-faenasImpoHabilitadas : int
-gatesHabilitados : List<Gate>
-semana : Date
-capacidad : Map<Integer, int[]> = new HashMap<Integer, int[]>()
-zonasHorarias : int
-gate : Gate
-serialVersionUID2 : long = 6795218864049653369L
-serialVersionUID5 : long = -6502828764596953737L
-nombre2 : String
-id3 : Integer
-serialVersionUID3 : long = 3697091875667879902L
-capacidadGate : CapacidadGate
-nombre : String
-id2 : Integer
-LOGGER : Logger = LoggerFactory
.getLogger(CapacidadGate.class)
-zonaHoraria : int
-gateModel : GateModel
-serialVersionUID6 : long = -1256189424264347777L
-idPlanificacion : Integer
-responsableModificacion : ResponsableModificacion
-numeroZonasHorariasSeleccionada : Integer
-capacidadGate2 : CapacidadGate
-zonasHorariasSelect : List<Integer>
-existeCapacidad : boolean = false
-requisitosVisualidadGate : boolean = true
-LOGGER2 : Logger = LoggerFactory
.getLogger(GateModel.class)
-idGateSeleccionado : int
-gatesListados : List<Gate> = new ArrayList<Gate>()
-criterioGateSeleccionado : CriterioGate
-nombreGateSeleccionado : String
-semanaSeleccionada : Date
-nombreConfiguracion : String
-modoEditable : boolean = false
-capacidad2 : Spread sheetModel
-fechaCapacidadGate : Date
-serialVersionUID7 : long = 452332891448775932L
-idCriterioGateSeleccionado : Integer
-criteriosGate : List<CriterioGate>
-id : Integer

```
+asignarSlots(diaSemana : int, zonaHoraria : int, slots : int) : void
+recuperarSlots(diaSemana : int, zonaHoraria : int) : int
+toString() : String
+CapacidadGate(zonasHorarias : int)
+isCapacidadGateDefinida() : boolean
+equals(obj : Object) : boolean
+hashCode() : int
+setMartes(slots : int) : void
+getMartes() : int
+getLunes() : int
+setViernes(slots : int) : void
+isModoEditable() : boolean
+setRequisitosVisualidadGate(requisitosVisualidadGate : Boolean) : void
+isExisteCapacidad() : boolean
+getRequisitosVisualidadGate() : Boolean
+getJueves() : int
+setSabado(slots : int) : void
+setJueves(slots : int) : void
+getViernes() : int
+getDomingo() : int
+ZonaHorariaModel(zonaHoraria : int, gateModel : GateModel)
+setLunes(slots : int) : void
+setMiercoles(slots : int) : void
+setDomingo(slots : int) : void
+getSabado() : int
+getMiercoles() : int
+getSlots() : int []
+ZonaHorariaModel()
+isTodosGatesHabilitados() : boolean
```

Anexo 5 – Clase Nave

Nave
-viajeNave2 : ViajeNave
-serialVersionUID4 : long = 476656289099173293L
-faenasUtilizadas : List<Faena> = new ArrayList<Faena>()
-faenasEliminadas : List<Faena> = new ArrayList<Faena>()
-faenasRegistradas : List<Faena> = new ArrayList<Faena>()
-idFaenaSeleccionado : Integer
-fechaTerminoFaena : Date
-serialVersionUID : long = 8329850820529265461L
-LOGGER : Logger = LoggerFactory .getLogger(FaenasNaveModel.class)
-faenaSeleccionada : Faena
-faenas : List<Faena>
-viajeNave : ViajeNave
-fechaInicioFaena : Date
-serialVersionUID2 : long = -5521513112714521031L
-viajeNaveSeleccionado : ViajeNave
-serialVersionUID3 : long = 1208506991860880609L
-serialVersionUID6 : long = -527859471839850497L
-fechaInicio : Date
-fechaTermino : Date
-viajeNave3 : ViajeNave
-semanaSeleccionadaDetalle : Date
-semanaSeleccionada2 : Date
-idFaena : Integer
-serialVersionUID5 : long = -4817079445146000229L
-navePuerto : NavePuerto
-tipoFaena : TipoFaena
-viajesNaves : List<ViajeNave>
-operador : String
-id : Integer
-servicio : String
-viaje : String
-nombreNave : String
-fechaArribo : Date
-semanaSeleccionada : Date
-faenasSeleccionadas : List<FaenaNave> = new ArrayList<FaenaNave>()
-faenasAsociadas : List<FaenaNave>
-obtenerFaenaSeleccionada() : Faena
+toString() : String
+getTipoFaenaSeleccionada() : TipoFaena
+eliminarFaenaSeleccionada() : void

Anexo 6 – Clase Faena

Faena
<pre> <<Constant>> -NAVE = "Nave" -friendlyName : String <<Constant>> -TIPO_CAMION = "Tipo Camión" <<Constant>> -NUMERO_BULTOS = "Número Bulto" <<Constant>> -TIPO_BULTO = "Tipo Bulto" -friendlyName2 : String <<Constant>> -ENTREGA = "Exportación" <<Constant>> -RETIRO = "Importación" <<Constant>> -DOCUMENTO_NAVIERO = "Documento Naviero(Booking/BL)" <<Constant>> -TIPO_CARGA = "Tipo Carga" <<Constant>> -OPERADOR_NAVIERO = "Operador Naviero" <<Constant>> -FECHA_ARRIBO_NAVE = "Fecha de Arribo de Nave" -atributosSeleccionados : List<String> -atributosDisponibles : List<AtributoTipoFaena> -modalidad : ModalidadTipoFaena -nombreConfiguracion : String -idPlanificacion : Integer -id : Integer -seriaVersionUID2 : long = -2405337085732418777L -configuracionesFaena : List<ConfiguracionFaena> -especificarNave : boolean = false -modalidades : List<ModalidadTipoFaena> -seriaVersionUID : long = -4754029346491611961L -glosaTipoFaena : String -seriaVersionUID3 : long = -321838592972009233L -numeroZonasHorarias : int -glosa : String -id3 : Integer -especificarNave2 : boolean -atributos : List<AtributoTipoFaena> -modalidad2 : ModalidadTipoFaena -seriaVersionUID4 : long = -1331793198044494484L -id2 : Integer -slotsFaena : Map<Integer, int[]> -tipoFaena : TipoFaena -idTipoFaena : Integer <<Constant>> -PESO = "Peso" <<Constant>> -DOCUMENTO_ADUANERO = "Documento Aduanero(DUS/DIN)" -tipoFaena2 : TipoFaena -nombre : String -duracion : int -seriaVersionUID6 : long = 8850213821472973980L </pre>

```

-zonaHoraria : int
-fecha : Date
<<Constant>> -NUMERO_CONTENEDOR =
"Número Contenedor"
-seria/VersionUID5 : long = 2285293358624566793L
-tipoFaenaSeleccionada : TipoFaena
-tiposFaenas : List<TipoFaena>
-idCriterioGlobalSeleccionado : Integer
-criteriosGlobalFaena : List<CriterioGlobalFaena>
-tiposFaenas2 : List<TipoFaena>
-gates : List<Gate>
-tipoFaenaSeleccionada2 : TipoFaena
-idTipoFaenaSeleccionada : int
-modoVer : boolean = false
-valorFijo : int
-porcentajeCapacidadGate : int
-seria/VersionUID8 : long = -2064965563034155825L
-criterio : TipoCriterioInicializacionSlotsFaena
-id4 : Integer
-modoEdicion : boolean = false
-slotsFaenas : Map<Integer, SpreadsheetModel> = null
-nombreCriterioGlobal : String
-mostrarTablaFaenas : boolean = false
-valoresInicializacionSlotsFaenas : List<ValorInicializacionSlotsFaena>
-faenaSeleccionada : Faena
-LOGGER : Logger = LoggerFactory
.getLogger(FaenasModel.class)
-seria/VersionUID7 : long = 6336443779333847954L
-faenasHabilitadas : boolean = false
-navePuertosSeleccionados : NavePuerto[]
-navePuertos : List<NavePuerto>
-planificacionSemanal : PlanificacionSemanal
-semanaSeleccionada : Date
-idGateSel : Integer
-idValorInicializacionSel : int
-modificacionFaenas : boolean
-friendlyName3 : String
<<Constant>> -VALOR_FIJO =
"Valor Fijo"
<<Constant>> -PORCENTAJE_CAPACIDAD_GATE = "Porcentaje Capacidad Gate"
-idPlanificacion2 : Integer
-id5 : Integer
-responsableModificacion : ResponsableModificacion
-tipoFaena3 : TipoFaena
-navePuerto : List<NavePuerto> = new ArrayList<NavePuerto>()
-slotsFaena2 : Map<Integer, int[]> = new HashMap<Integer, int[]>()
-seria/VersionUID9 : long = -3351672352596819090L
-semana : Date
-gate : Gate

```

```

-AtributoTipoFaena(friendlyName : String)
-ModalidadTipoFaena(friendlyName : String)
+TipoFaenaModel(tipoFaena : TipoFaena)
+TipoFaenaModel()
+isEspecificaNave() : boolean
+isEspecificaNave2() : boolean
+toString() : String
+hashCode() : int
+equals(obj : Object) : boolean
+isModoEdicion() : boolean
+isMostrarTablaFaenas() : boolean
+isFaenasHabilitadas() : boolean
+getNumeroSlots() : String
+FaenasModel()
+friendlyName2() : String
+getTipoFaena(idTipoFaena : Integer) : TipoFaena
+getValor() : int
+getTipoCriterio() : TipoCriterioInicializacionSlotsFaena
+toString2() : String
+ValorInicializacionSlotsFaena(criterio : TipoCriterioInicializacionSlotsFaena, valor : int)
+getModoVisualizacion() : boolean
+isModoVer() : boolean
+getSlotsFaenaByID(id : int) : SpreadsheetModel
+isModificacionFaenas() : boolean
+getValorInicializacionSeleccionado() : ValorInicializacionSlotsFaena
+requisitosCumplidosHabilitacionFaenas() : boolean
-TipoCriterioInicializacionSlotsFaena(friendlyName : String)
+friendlyName3() : String
+slotsDefinidos() : boolean
+existenCriteriosGlobales() : boolean
+equals2(obj : Object) : boolean
+getZonasHorarias() : int
+hashCode2() : int
+friendlyName() : String

```

Anexo 7 – Clase Controlador

Controlador
<pre> -LOGGER2 : Logger = LoggerFactory .getLogger(UsuariosController.class) -usuarioService2 : UsuarioService -messagesController : MessagesController -ubicacionService : UbicacionService -serialVersionUID : long = 5851826683832488533L -perfilUsuario : PerfilUsuario -authorities : ArrayList<GrantedAuthority> = new ArrayList<GrantedAuthority>() -LOGGER : Logger = LoggerFactory .getLogger(OpenGateUserDetailsServiceImpl.class) -usuarioService : UsuarioService -LOGGER3 : Logger = LoggerFactory .getLogger(DefaultUsuarioService.class) -validator : Validator -jdbcTemplate : JdbcTemplate -namedJT : NamedParameterJdbcTemplate -messageSource : MessageSource -ubicacionService2 : UbicacionService -LOGGER4 : Logger = LoggerFactory .getLogger(CamionesController.class) -transportistaService : TransportistaService -messagesController2 : MessagesController -LOGGER5 : Logger = LoggerFactory .getLogger(DefaultTransportistaService.class) -jdbcTemplate2 : JdbcTemplate -namedJT2 : NamedParameterJdbcTemplate -messageSource2 : MessageSource -validator2 : Validator -LOGGER6 : Logger = LoggerFactory .getLogger(ConductoresController.class) -transportistaService2 : TransportistaService -messagesController3 : MessagesController -LOGGER7 : Logger = LoggerFactory .getLogger(TransportistasController.class) -messagesController4 : MessagesController -transportistaService3 : TransportistaService -LOGGER8 : Logger = LoggerFactory .getLogger(FaenasController.class) -faenaService : FaenaService -planificacionService : PlanificacionService -gateService : GateService -reservaService : ReservaService -messagesController5 : MessagesController -datosSesionController : DatosSesionController -fechasController : FechasController -usuarioService3 : UsuarioService -LOGGER9 : Logger = LoggerFactory .getLogger(DefaultFaenaService.class) -naveService : NaveService -gateService2 : GateService -jdbcTemplate3 : JdbcTemplate </pre>

```

-namedJT3 : NamedParameterJdbcTemplate
-messageSource3 : MessageSource
-validator3 : Validator
-usuarioService4 : UsuarioService
-LOGGER23 : Logger = LoggerFactory
.getLogger(DefaultContenedorService.class)
-----
-mensajesSql : MessageSource
-jdbcTemplate10 : JdbcTemplate
-LOGGER10 : Logger = LoggerFactory
.getLogger(DefaultGateService.class)
-----
-validator4 : Validator
-jdbcTemplate4 : JdbcTemplate
-namedJT4 : NamedParameterJdbcTemplate
-messageSource4 : MessageSource
-usuarioService5 : UsuarioService
-LOGGER11 : Logger = LoggerFactory
.getLogger(ReporteSemanaUsuarioTerminalBuilder.class)
-----
-RES_PATH : String = "/reportes/reportesSemana/"
-gateService3 : GateService
~reporte1DirectoryName : String = "ingreso_camiones_tiempo_estadia_terminal/"
~reporte2DirectoryName : String = "rango_estadia_camiones_con_reserva/"
~reporte3DirectoryName : String = "rango_estadia_camiones_sin_reserva/"
~reporte5DirectoryName : String = "utilizacion_reservas_por_zona_horaria/"
-fillMgr : JasperFillManager
-jasperContext : JasperReportsContext
-jdbcTemplate5 : JdbcTemplate
-resourcesPath : String
-LOGGER22 : Logger = LoggerFactory
.getLogger(DefaultReservaService.class)
-----
-validator7 : Validator
-faenaService7 : FaenaService
-bultoService2 : BultoService
-naveService4 : NaveService
-transportistaService5 : TransportistaService
-contenedorService2 : ContenedorService
-jdbcTemplate9 : JdbcTemplate
-namedJT6 : NamedParameterJdbcTemplate
-messageSource6 : MessageSource
-semana : DateTime
-gateService4 : GateService
-LOGGER12 : Logger = LoggerFactory

```

```

.getLogger(DefaultPlanificacionService.class)
-gateService5 : GateService
-faenaService2 : FaenaService
-jdbcTemplate6 : JdbcTemplate
-messageSource5 : MessageSource
-validator5 : Validator
-LOGGER13 : Logger = LoggerFactory
.getLogger(PlanificacionController.class)
-messagesController6 : MessagesController
-planificacionService2 : PlanificacionService
-faenasController : FaenasController
-LOGGER14 : Logger = LoggerFactory
.getLogger(DefaultReporteService.class)
-jdbcTemplate7 : JdbcTemplate
~context : ApplicationContext
-LOGGER15 : Logger = LoggerFactory
.getLogger(ReservasController.class)
-NOMBRE_RESERVA_MODEL : String = "reservaModel"
-faenaService3 : FaenaService
-bultoService : BultoService
-reservaService2 : ReservaService
-gateService6 : GateService
-contenedorService : ContenedorService
-naveService2 : NaveService
-usuarioService6 : UsuarioService
-transportistaService4 : TransportistaService
-planificacionService3 : PlanificacionService
-datosSesionController2 : DatosSesionController
-messagesController7 : MessagesController
-LOGGER16 : Logger = LoggerFactory
.getLogger(DatosSesionController.class)
-LOGGER17 : Logger = LoggerFactory
.getLogger(GateController.class)
-LOGGER21 : Logger = LoggerFactory
.getLogger(DefaultNaveService.class)
-jdbcTemplate8 : JdbcTemplate
-messageSourceSql : MessageSource
-faenaService6 : FaenaService
-validator6 : Validator
-namedJT5 : NamedParameterJdbcTemplate
-LOGGER20 : Logger = LoggerFactory
.getLogger(NavesController.class)
-naveService3 : NaveService
-faenaService5 : FaenaService
-messagesController10 : MessagesController
-datosSesionController6 : DatosSesionController
+NUMERO_SEMANAS : int = 5
-gateService7 : GateService
-messagesController8 : MessagesController
-datosSesionController3 : DatosSesionController
-usuarioService7 : UsuarioService
-totalGates : int

```

```

-LOGGER18 : Logger = LoggerFactory
.getLogger(ReportesController.class)
-reporteService : ReporteService
-datosSesionController4 : DatosSesionController
-LOGGER19 : Logger = LoggerFactory
.getLogger(TiposFaenasController.class)
-faenaService4 : FaenaService
-messagesController9 : MessagesController
-datosSesionController5 : DatosSesionController

+loadUserByUseName(username : String) : UserDetails
+registrarPlanificadorGate(planificadorGate : PlanificadorGate) : void
+recuperarPlanificadorGate(idPlanificador : int) : PlanificadorGate
+recuperarPerfilUsuario(nombreUsuario : String) : PerfilUsuario
+registrarUsuarioExterno(usuarioExterno : UsuarioExterno) : void
+recuperarUsuarioExterno(idUsuarioExterno : int) : UsuarioExterno
+aceptarSolicitudUsuario(idPersona : int) : void
+listarUsuariosExternosEstado(estadoPersona : EstadoPersona) : List<UsuarioExterno>
+getUsuariosExternosPendientesModel() : UsuariosExternosModel
+getUsuariosExternosAceptadosModel() : UsuariosExternosModel
+aceptarSolicitudUsuario(usuariosExternosModel : UsuariosExternosModel) : boolean
+getPlanificadorModel() : PlanificadorModel
+getUsuarioExternoModel() : UsuarioExternoModel
+init() : void
+registrarPlanificadorGate2(planificadorGate : PlanificadorGate) : void
-validarBeanPlanificador(planificadorGate : PlanificadorGate) : void
-validarPlanificador(planificadorGate : PlanificadorGate) : Set<ConstraintViolation<PlanificadorGate>>
-insertarRolPersona(idPersona : int, roles : List<TipoRol>) : void
-insertarRol(idPlanificador : Integer, tipoRol : TipoRol) : void
-insertarUsuario(perfilUsuario : PerfilUsuario, idPersona : int) : void
-nombreUsuarioRegistrado(perfilUsuario : PerfilUsuario) : boolean
-insertarPlanificador(idPersona : int, planificadorGate : PlanificadorGate) : void
+recuperarPlanificadorGate2(idPlanificador : int) : PlanificadorGate
-recuperarInformacionPlanificador(idPlanificador : int) : PlanificadorGate
-recuperarRolesUsuario(idPlanificador : int) : List<TipoRol>
-recuperarPerfilUsuario(idPlanificador : int) : PerfilUsuario
+recuperarPerfilUsuario2(nombreUsuario : String) : PerfilUsuario
-obtenerTipoPersona(nombreUsuario : String) : TipoPersona
+registrarUsuarioExterno2(usuarioExterno : UsuarioExterno) : void
-insertarUsuarioExterno(idPersona : int, usuarioExterno : UsuarioExterno) : void
-obtenerRut(rut : String) : String []
-insertarPersona(tipoPersona : TipoPersona, estadoPersona : EstadoPersona) : int
-validarBeanUsuarioExterno(usuarioExterno : UsuarioExterno) : void

```

```

-validarUsuarioExterno(usuarioExterno : UsuarioExterno) : Set<ConstraintViolation<UsuarioExterno>>
+recuperarUsuarioExterno2(idUsuarioExterno : int) : UsuarioExterno
-recuperarInformacionUsuarioExterno(idUsuarioExterno : int) : UsuarioExterno
+init2() : void
+getResultadosList() : Integer []
+getTipoFaena(cdTipoFaena : int) : TipoFaena
+getTipoFaenaModel() : TipoFaenaModel
+getTipoFaenaModel(cdTipoFaena : int) : TipoFaenaModel
+getTiposFaenasModel() : TiposFaenasModel
+registrarTipoFaena(tipoFaenaModel : TipoFaenaModel) : boolean
-generarTipoFaena(model : TipoFaenaModel) : TipoFaena
+getTotalRegistros() : int
+seleccionarTipoFaena(model : TiposFaenasModel, tipoFaena : TipoFaena) : void
+listarTransportistas() : List<Transportista>
+recuperarTransportista(idTransportista : int) : Transportista
+listarCamiones(idTransportista : int) : List<Camion>
+recuperarCamion(idCamion : int) : Camion
+getReportesModel() : ReportesModel
+generarReporteSemanal(model : ReportesModel) : StreamedContent
+listarConductores(idTransportista : int) : List<Conductor>
+recuperarConductor(idConductor : int) : Conductor
+registrarConductor(conductor : Conductor) : Integer
-validateConductor(conductor : Conductor) : void
+registrarCamion(camion : Camion) : Integer
-validateCamion(camion : Camion) : void
+registrarTransportista(transportista : Transportista) : Integer
-validateTransportista(transportista : Transportista) : void
+aceptarSolicitudUsuario2(idPersona : int) : void
-modificarEstadoPersona(idPersona : int, estadoPersona : EstadoPersona) : void
+getCamionModel() : CamionModel
+registrarCamion(model : CamionModel) : boolean
-generarCamion(model : CamionModel) : Camion
-obtenerEstadoActual(idPersona : int) : EstadoPersona
+listarUsuariosExternosEstado2(estadoPersona : EstadoPersona) : List<UsuarioExterno>
+recuperarNombreUsuario2(idUsuarioExterno : Integer) : String
+solicitarRegistroUsuarioExterno(usuarioExternoModel : UsuarioExternoModel) : boolean
+OpenGateUserDetails(perfilUsuario : PerfilUsuario)
+getAuthorities() : Collection<GrantedAuthority>
+listarTransportistas2() : List<Transportista>
+recuperarTransportista2(idTransportista : int) : Transportista
+listarCamiones2(idTransportista : int) : List<Camion>
+recuperarCamion2(idCamion : int) : Camion
+getReporteSemanal(usuario : String, tipoRol : TipoRol, dateTime : DateTime) : StreamedContent
+registrarCamion2(camion : Camion) : Integer
+listarConductores2(idTransportista : int) : List<Conductor>
+getConductoresModel(idTransportista : Integer) : ConductoresModel
+getConductorModel() : ConductorModel
+recuperarConductor(idConductor : int) : ConductorModel
+registrarConductor(model : ConductorModel) : boolean
-generarConductor(model : ConductorModel) : Conductor
+recuperarConductor2(idConductor : int) : Conductor

```

```

+registrarConductor2(conductor : Conductor) : Integer
+registrarTransportista2(transportista : Transportista) : Integer
+getPassword() : String
+getFaenasModel() : FaenasModel
+cargarFaenas(model : FaenasModel, modoEdicion : boolean) : void
-filtrarFaenasSinSlots(model : FaenasModel) : void
+poblarFaenasModelo(model : FaenasModel) : void
+listarCriteriosInicializacionSlots() : List<ValorInicializacionSlotsFaena>
+inicializarVistaHabilitacionFaenas(model : FaenasModel) : void
+inicializarFaenasModelo(model : FaenasModel) : void
-obtenerSlotsIniciales(valorInicializacionSeleccionado : ValorInicializacionSlotsFaena, zonaHoraria : int, dia : int, capacida...
+guardarFaenas(model : FaenasModel) : void
-generarListaFaenas(model : FaenasModel) : List<Faena>
+getTimeZone(date : Date) : TimeZone
+getTimeZone() : TimeZone
+getTimeZoneID() : TimeZone
+getPerfilUsuario() : PerfilUsuario
+usuarioTieneRol(rol : TipoRol) : boolean
+obtenerNombreUsuario() : String
+getNumeroZonasHorarias() : SelectItem []
+generarExcelCapacidadGate(model : GateModel) : StreamedContent
+getGateModel() : GateModel
+existenMultiplesGates() : boolean
+recuperarCapacidadGate(gateModel : GateModel, editable : boolean) : void
-generateSpreadsheetModel(capacidadGate : CapacidadGate) : SpreadsheetModel
+registrarCapacidadGateVacía(gateModel : GateModel) : boolean
+actualizarCapacidadGate(model : GateModel) : void
+inicializarGateModel(model : GateModel) : void
+guardarConfiguracionGate(gateModel : GateModel) : void
+registrarCapacidadConfiguracion(gateModel : GateModel) : void
-generarResponsableModificacion2() : ResponsableModificacion
+seleccionarCriterioGate(gateModel : GateModel) : void
+existeCapacidadGate(gateModel : GateModel) : void
+rangoFechaZonaHoraria2(zonaHoraria : int, zonasHorarias : int) : String
+obtenerNombrePerfil() : List<String>
+getReglamentoOpenGate() : StreamedContent
-generarResponsableModificacion() : ResponsableModificacion
+rangoFechaZonaHoraria(zonaHoraria : int, zonasHorarias : int) : String
+getSlotsAgregadosFaenas(model : FaenasModel) : List<ZonaHorariaFaena>
+spreadsheetModelAListaZonaHorariaFaena(model : SpreadsheetModel, capacidadGate : CapacidadGate) : List<ZonaHo...
+calcularPorcentajeUsoCapacidadGate(slotsHabilitados : int, capacidadGate : int) : String
+cargarHorarioDisponible(reservaModel : ReservaModel) : boolean
-calcularMinutosPorSlot(numeroZonasHorarias : int) : int

```

```

-obtenerSlotsUtilizados(reservasSemana : List<Reserva>, dia : int, zonaHoraria : int) : int
-inicializarScheduler(scheduleReservaModel : ScheduleReservaModel) : void
+seleccionarTipoFaena(faenasModel : FaenasModel) : void
+existeFaenasHabilitadas(faenasModel : FaenasModel) : void
+guardarCriterioFaenasGlobal(faenasModel : FaenasModel) : void
-obtenerConfiguracionesFaenasModelo(faenasModel : FaenasModel) : List<ConfiguracionFaena>
+habilitarFaenasCriterioGlobal(faenasModel : FaenasModel) : void
+listarTiposFaenas() : List<TipoFaena>
+recuperarTipoFaena(codigoTipoFaena : int) : TipoFaena
+registrarTipoFaena(tipoFaena : TipoFaena) : int
+habilitarTipoFaenaPlanificaciones(idTipoFaena : int, idResponsable : int) : void
+totalRegistrosTipoFaena() : int
+recuperarFaena(idFaena : int) : Faena
+recuperarFaena(idGate : int, semana : Date, idTipoFaena : int) : Faena
+registrarFaenas(faenas : List<Faena>) : void
+init3() : void
+listarTiposFaenas2() : List<TipoFaena>
+recuperarTipoFaena2(codigoTipoFaena : int) : TipoFaena
-recuperarAtributosTipoFaena(codigoTipoFaena : int) : List<AtributoTipoFaena>
+registrarTipoFaena2(tipoFaena : TipoFaena) : int
-registrarAtributosTipoFaena(idTipoFaena : Integer, atributos : List<AtributoTipoFaena>) : void
-validateTipoFaena(tipoFaena : TipoFaena) : void
+totalRegistrosTipoFaena2() : int
+recuperarFaena2(idFaena : int) : Faena
+listarFaenasPlanificacion2(idPlanificacion : int) : List<Faena>
-validateFaena(faena : Faena) : void
+listarFaenasPlanificacion2(semana : Date) : List<Faena>
-recuperarSlotsFaena(idFaena : Integer) : Map<Integer, int[]>
-recuperarZonasHorarias(idFaena : int) : int
+registrarFaenas2(faenas : List<Faena>) : void
-insertarFaena(idTipoFaena : Integer, idPlanificacion : Integer, idResponsable : Integer) : int
-guardarSlotsFaena(idFaena : int, slotsFaena : Map<Integer, int[]>, isModificacion : boolean) : void
+modificarFaenas2(faenas : List<Faena>) : void
-obtenerIDHistoria(idFaena : Integer) : int
+listarFaenasPlanificacion2(semana : Date, idGate : int) : List<Faena>
+existenFaenasHabilitadas2(idGate : Integer, semana : Date) : boolean
+registrarCriterioGlobalFaena2(criterioGlobalFaena : CriterioGlobalFaena) : void
-insertarConfiguracionFaena(criterioGlobalFaena : CriterioGlobalFaena, idCriterio : int) : void
-insertarConfiguracionFaena(configuracionFaena : ConfiguracionFaena, idCriterioTipo : int) : void
-insertarCriterioTipoFaena(configuracionFaena : ConfiguracionFaena, idCriterio : int) : int
-insertarCriterioGlobal(criterioGlobalFaena : CriterioGlobalFaena) : int
-validateBeanCriterioGlobalFaena(criterioGlobalFaena : CriterioGlobalFaena) : void
-validateCriterioGlobalFaena(criterioGlobalFaena : CriterioGlobalFaena) : Set<ConstraintViolation<CriterioGlobalFaena>>
+recuperarCriterioGlobalFaena2(idCriterioGlobal : int) : CriterioGlobalFaena
-obtenerConfiguracionesFaenas(idCriterioGlobal : int) : List<ConfiguracionFaena>
-obtenerZonasHorarias(idCriterioTipo : Integer) : int
-obtenerSlotsFaena(idCriterioTipo : Integer, zonasHorarias : int) : Map<Integer, int[]>
+registrarFaenasCriterioGlobal2(idCriterioGlobal : int, idGate : int, semana : Date, idResponsable : int) : void
+init4() : void
+modificarCapacidadGate(idGate : int, capacidadGate : int [], semana : Date, idResponsable : int) : void
+recuperarCapacidadGate(idGate : int, semana : Date) : CapacidadGate
+recuperarResponsableModificacion(idHistoria : int) : ResponsableModificacion

```

```

-obtenerIDHistoria(idGate : int, semana : Date) : int
#obtenerZonasHorarias(idGate : int, semana : Date) : int
+registrarCapacidadGate(idGate : int, capacidadGate : CapacidadGate, semana : Date) : Integer
+recuperarIDPlanificacion(idGate : int, semana : Date) : Integer
-registrarPlanificacionSemana(idGate : int, semanaRecuperar : SqlParameterValue, idHistoriaModificacion : int) : int
-validarCapacidadGate(capacidadGate : CapacidadGate, grupo : Class<?>) : Set<ConstraintViolation<CapacidadGate>>
+listarGates() : List<Gate>
+modificarCapacidadGateZonaHoraria(idGate : int, semana : Date, zonaHoraria : int, capacidades : int [], idResponsable : i..
+registrarGate(gate : Gate) : void
+registrarCriterioGate(idPlanificacion : int, nombreConfiguracion : String) : void
+insertarConfiguracionGate(idCriterio : int, capacidadGate : CapacidadGate) : void
+recuperarCriterioGate(idCriterio : int) : CriterioGate
#recuperarConfiguracionGate(idCriterio : int) : CapacidadGate
-completarCapacidadGateCriterio(idCriterio : int, capacidadGate : CapacidadGate) : void
+recuperarGate(idGate : int) : Gate
-obtenerZonasHorariasConfiguracion(idCriterio : int) : int
+listarCriteriosGate() : List<CriterioGate>
+recuperarCapacidadGate(idPlanificacion : int) : CapacidadGate
+existeCapacidadGate(idGate : int, semana : Date) : Boolean
+registrarHistoriaModificacion(idResponsable : int) : Integer
+actualizarHistoriaModificacion(idHistoria : int, idResponsable : int) : void
+recuperarGatesPlanificados(semana : Date) : Map<Gate, Boolean>
-obtenerFaenaVacía(tipoFaena : TipoFaena, gate : Gate, idPlanificacion : Integer, semana : Date, zonasHorarias : int) : Fa...
+listarCriteriosGlobalFaenas2() : List<CriterioGlobalFaena>
+habilitarTipoFaenaPlanificaciones2(idTipoFaena : int, idResponsable : int) : void
-obtenerSlotsVacíos(zonasHorarias : int) : Map<Integer, int[]>
-obtenerZonasHorariasPlanificacion(idPlanificacion : Integer) : int
-planificacionPoseeTipoFaena(idPlanificacion : Integer, idTipoFaena : Integer) : boolean
+recuperarFaena2(idGate : int, semana : Date, idTipoFaena : int) : Faena
+recuperarNumeroFaenasHabilitadas2(idGate : Integer, semana : Date) : int
+modificarFaenas(faenas : List<Faena>) : void
+listarFaenasPlanificacion(idPlanificacion : int) : List<Faena>
+listarFaenasPlanificacion(semana : Date) : List<Faena>
+listarFaenasPlanificacion(semana : Date, idGate : int) : List<Faena>
+existenFaenasHabilitadas(idGate : Integer, semana : Date) : boolean
+registrarCriterioGlobalFaena(criterioGlobalFaena : CriterioGlobalFaena) : void

```

```

+recuperarCriterioGlobalFaena(idCriterioGlobal : int) : CriterioGlobalFaena
+registrarFaenasCriterioInicial(idCriterioGlobal : int, idGate : int, semana : Date, idResponsable : int) : void
+listarCriteriosGlobalFaenas() : List<CriterioGlobalFaena>
+recuperarNumeroFaenasHabilitadas(idGate : Integer, semana : Date) : int
+existeCapacidadGate(faenasModel : FaenasModel) : boolean
+getUsername() : String
+isAccountNonExpired() : boolean
+isAccountNonLocked() : boolean
+isCredentialsNonExpired() : boolean
+isEnabled() : boolean
-generarObjetoUsuarioExterno(usuarioExternoModel : UsuarioExternoModel) : UsuarioExterno
+ReporteSemanalUsuarioTerminalBuilder()
#getJasperPrintList() : List<JasperPrint>
-setParametrosIndice(parametrosIndice : Map<String, Object>, gatesPlanificados : int) : void
-getNumGatesPlanificados(gates : Map<Gate, Boolean>) : int
-generarReporteIngresosTerminal(gate : Gate, semana : DateTime, tienePlanificacion : Boolean, numeroPagina : int) : Jas...
#getLogger() : Logger
+registrarPlanificadorGate(planificadorModel : PlanificadorModel) : boolean
+getTransportistaModel() : TransportistaModel
+recuperarCapacidadGate2(idGate : int, semana : Date) : CapacidadGate
+recuperarCapacidadGate2(idPlanificacion : int) : CapacidadGate
+registrarCapacidadGate(idGate : int, capacidad : CapacidadGate, semana : Date) : Integer
+modificarCapacidadGate2(idGate : int, capacidadGate : int []], semana : Date, idResponsable : int) : void
+listarGates2() : List<Gate>
+modificarCapacidadGateZonaHoraria2(idGate : int, semana : Date, zonaHoraria : int, capacidades : int [], idResponsable ...
+registrarGate2(gate : Gate) : void
+registrarCriterioGate2(idPlanificacion : int, nombreConfiguracion : String) : void
+recuperarCriterioGate2(idCriterio : int) : CriterioGate
+listarCriteriosGate2() : List<CriterioGate>
+existeCapacidadGate2(idGate : int, semana : Date) : Boolean
+recuperarGate2(idGate : int) : Gate
+recuperarIDPlanificacion2(idGate : int, semana : Date) : Integer
+recuperarGatesPlanificados2(semana : Date) : Map<Gate, Boolean>
+registrarHistoriaModificacion2(idResponsable : int) : Integer
+recuperarResponsableModificacion2(idHistoria : int) : ResponsableModificacion
+cargarPlanificacion(model : FaenasModel) : void
+getSlotsHighlightStyle(slotsUsadosZonaHoraria : int, slotsCapacidadZonaHoraria : int) : String
+getReporteSemanal(usuario : String, tipoRol : TipoRol, semana : DateTime) : StreamedContent
+actualizarFaenas(reservaModel : ReservaModel) : void
-actualizarIndicadoresAtributos(reservaModel : ReservaModel, faena : Faena) : void
+actualizarTransporte(manifiestoTransporteModel : ManifiestoTransporteModel) : void
+actualizarTransporte(reservaModel : ReservaModel) : void
+calcularFechaReserva(semana : Date, dia : int) : Date
-calcularMinutosPorSlot2(numeroZonasHorarias : int) : int
-calcularTotalSlotsHabilitados(faenasHabilitadas : List<Faena>, dia : int, zonaHoraria : int) : int
+cancelarReserva(reserva : Reserva) : void
+cargarHorarioDisponible(reservaHorarioModel : ReservaHorarioModel) : void
+cargarHorarioDisponible(reservaModel : ReservaModel) : void
+cargarReservas(model : ReservasModel) : void
+generarReservasDia(reservasModel : ReservasModel) : boolean
-contarReservas(reservas : List<Reserva>) : int []
+cargarReservasTransportista(reservasModel : ReservasUsuarioBasicoModel) : void
+cargarReservasUsuarioBasico(model : ReservasUsuarioBasicoModel) : void
+cargarUsuariosExternos(reservaModel : ReservaModel) : void

```

```

+faenaSinNavesConAtributoNave(reservaModel : ReservaModel) : boolean
-filtrarFaenas(faenasSemana : List<Faena>, modalidad : ModalidadTipoFaena) : List<Faena>
+formatearFecha(fecha : Date) : String
-generarObjetoReserva(reservaModel : ReservaModel) : Reserva
+generarReservasSemanaGate(model : ReservasModel) : boolean
+getDetalleReservaModel(idReserva : int) : DetalleReservaModel
+getFechaReserva(reserva : Reserva) : String
+getManifiestoTransporteModel(idReserva : int) : ManifiestoTransporteModel
+getModalidades() : ModalidadTipoFaena []
+getNombreFechaSeleccionada(reservaModel : ReservaModel) : String
+getReservaHorarioModel(idReserva : int) : ReservaHorarioModel
+getReservaModel() : ReservaModel
+getReservasModel() : ReservasModel
+getReservasUsuarioBasicoModel() : ReservasUsuarioBasicoModel
+getTiposCarga() : TipoCarga []
-inicializarScheduler2(scheduleReservaModel : ScheduleReservaModel) : void
+isNaveDisponible(reservaModel : ReservaModel, dia : Integer, zonaHoraria : Integer) : boolean
+isNaveDisponibleHorario(reservaHorarioModel : ReservaHorarioModel, dia : Integer, zonaHoraria : Integer) : boolean
+modificarHorarioReserva(reservaHorarioModel : ReservaHorarioModel) : boolean
+modificarManifiestoTransporte(manifiestoTransporteModel : ManifiestoTransporteModel) : boolean
-obtenerNavePuerto(navesPuerto : List<NavePuerto>, viajeNave : ViajeNave) : NavePuerto
-obtenerSlotsUtilizados2(reservasSemana : List<Reserva>, dia : int, zonaHoraria : int) : int
+permiteEdicionAvanzada(idReserva : int) : Boolean
+permiteEdicionTransportista(idReserva : int) : boolean
+registrarReserva(reservaModel : ReservaModel) : boolean
+requiereInformacionTransportista(reservaModel : ReservaModel) : void
+actualizarTransportistas(model : ReservaModel, idTransportistaSeleccionado : Integer) : void
+actualizarConductores(model : ReservaModel) : void
+actualizarConductores(model : ReservaModel, idConductorSeleccionado : Integer) : void
+actualizarCamiones(model : ReservaModel) : void
+actualizarCamiones(model : ReservaModel, idCamionSeleccionado : Integer) : void
+reservaHorarioPasado(detalleReservaModel : DetalleReservaModel) : boolean
-resetearAtributosModelo(reservaModel : ReservaModel) : void
-resetearModelo(reservaModel : ReservaModel) : void
+resetearScheduler(reservaModel : ReservaModel) : void
+seleccionarCamion(manifiestoTransporteModel : ManifiestoTransporteModel) : void
+seleccionarCamion(reservaModel : ReservaModel) : void
+seleccionarConductor(manifiestoTransporteModel : ManifiestoTransporteModel) : void
+seleccionarConductor(reservaModel : ReservaModel) : void
+seleccionarFaena(reservaModel : ReservaModel) : void
+seleccionarHorario(reservaModel : ReservaModel, dia : Integer, zonaHoraria : Integer) : void
+seleccionarHorarioNuevo(reservaHorarioModel : ReservaHorarioModel, dia : Integer, zonaHoraria : Integer) : void
+seleccionarReserva(reserva : Reserva, reservasDia : ReservasDia) : void

```

```

-transformarDiaZonaAFecha(semanaSeleccionada : Date, dia : Integer, zonaHoraria : Integer, numeroZonasHorarias : Inte...
+validarContenedor(reservaModel : ReservaModel) : void
+validarPasosReserva(evento : FlowEvent) : String
+obtenerNombreUsuarioReserva(reserva : Reserva) : String
+recuperarRepresentanteReserva(reserva : Reserva) : PlanificadorGate
+filtrarReservas(reservasDia : ReservasDia) : void
+listarResumenesPlanificacionesSemanales(semanaSeleccionada : Date) : List<ResumenPlanificacionSemana>
+actualizarHistoriaModificacion2(idHistoria : int, idResponsable : int) : void
+registrarTransportista(model : TransportistaModel) : boolean
+AbstractReporteBuilder()
+build() : StreamedContent
+init5() : void
+recuperarPlanificacionSemana1(idGate : int, semana : Date) : PlanificacionSemana1
+recuperarPlanificacionSemana2(idGate : int, semana : Date) : PlanificacionSemana2
+listarResumenesPlanificacionesSemanales2(semana : Date, numeroSemanasAnteriores : int, numeroSemanasSiguiente...
-validatePlanificacionSemana1(planificacion : PlanificacionSemana1) : void
+listarResumenesPlanificacionesSemanales(semana : Date, numeroSemanasAnteriores : int, numeroSemanasSiguientes ...
-validateResumenPlanificacionSemana(resumenPlanificacion : ResumenPlanificacionSemana) : void
#getJasperPrintList2() : List<JasperPrint>
#getLogger2() : Logger
#getReporte(reporte : String, parametros : Map<String, Object>) : JasperPrint
+FaenaRowMapper()
+FaenaRowMapper(gateService : GateService)
+mapRow(rs : ResultSet, rowNum : int) : Faena
#getResourceStream(filename : String) : InputStream
-generarTransportista(model : TransportistaModel) : Transportista
-generarObjetoPlanificador(planificadorModel : PlanificadorModel) : PlanificadorGate
+cargarCoordenadas(usuarioExternoModel : UsuarioExternoModel) : void
+recuperarNombreUsuario(idUsuarioExterno : Integer) : String
+asignarFaena(faenasNaveModel : FaenasNaveModel) : void
+asignarFaenasNave(faenasNaveModel : FaenasNaveModel) : boolean
-faenaYaRegistrada(faenaNave : FaenaNave, faenasRegistradas : List<Faena>) : boolean
+getFaenasNaveModel(navesModel : NavesModel) : FaenasNaveModel
-filtrarFaenasParaNave(listarFaenasPlanificacion : List<Faena>) : List<Faena>
-obtenerIndiceFaenaAsociada(faena : Faena, faenasListadas : List<Faena>) : Integer
+getFechaFinSemana(fechaArribo : Date) : Date
+getNavePuertoModel(navesModel : NavesModel) : NavePuertoModel
+getNavesModel() : NavesModel
+obtenerFechaMaximaFaena(faenasNaveModel : FaenasNaveModel) : Date
+obtenerFechaMinimaFaena(faenasNaveModel : FaenasNaveModel) : Date
-obtenerNavePuerto(idViajeNave : Integer, navesPuertos : List<NavePuerto>) : NavePuerto
+comprobarFaenasSemana(navesModel : NavesModel) : void
+seleccionarNave(navesModel : NavesModel, viajeNave : ViajeNave) : void
+tieneFaenasAsociadas(nave : ViajeNave) : boolean
+totalRegistros() : int
+quitarFaena(faenaNave : FaenaNave, faenasNaveModel : FaenasNaveModel) : void
-obtenerFaenaUtilizada(idFaena : Integer, faenasUtilizadas : List<Faena>) : Faena
-obtenerFaenaRegistrada(idFaena : Integer, faenasRegistradas : List<Faena>) : Faena
+seleccionarFaena(faenasNaveModel : FaenasNaveModel) : void
+init6() : void
+listarViajesNaves() : List<ViajeNave>
+recuperarViajeNave(idViajeNave : int) : ViajeNave
+recuperarNavePuerto(idFaena : int, idViajeNave : int) : NavePuerto
+recuperarNavesAsociadas(idFaena : int) : List<NavePuerto>
+registrarNavePuerto(idFaena : int, navePuerto : NavePuerto) : void
+totalNavesRegistradas() : int

```

```

-validarBeanNavePuerto(navePuerto : NavePuerto) : void
-validarNavePuerto(navePuerto : NavePuerto) : Set<ConstraintViolation<NavePuerto>>
+recuperarFaenasAsociadas(idViajeNave : int) : List<Faena>
+eliminarFaenaAsociada(idFaena : int, viajeNave : ViajeNave) : void
+recuperarFaenasAsociadasSemana(idViajeNave : int, semana : Date) : List<Faena>
+listarViajesNaves2() : List<ViajeNave>
+recuperarFaenasAsociadas2(idViajeNave : int) : List<Faena>
+recuperarFaenasAsociadasSemana2(idViajeNave : int, semana : Date) : List<Faena>
+recuperarNavePuerto2(idFaena : int, idViajeNave : int) : NavePuerto
+recuperarNavesAsociadas2(idFaena : int) : List<NavePuerto>
+registrarNavePuerto2(idFaena : int, navePuerto : NavePuerto) : void
+eliminarFaenaAsociada2(idFaena : int, viajeNave : ViajeNave) : void
+totalNavesRegistradas2() : int
+recuperarViajeNave2(idViajeNave : int) : ViajeNave
+registrarReserva(reserva : Reserva) : Reserva
+modificarHorarioReserva(reserva : Reserva, dia : Integer, zonaHoraria : Integer) : void
+listarReservas(semana : Date, idReservador : Integer) : List<Reserva>
+listarReservasAsociadasFaena(idFaena : int) : List<Reserva>
+listarReservas(semana : Date) : List<Reserva>
+recuperarReserva(idReserva : int) : Reserva
+aprobarReserva(idReserva : int) : void
+cancelarReserva(idReserva : int) : void
+registrarManifiestoTransporte(manifiestoTransporte : ManifiestoTransporte) : void
+recuperarManifiestoTransporte(idReserva : int) : ManifiestoTransporte
+modificarManifiestoTransporte(manifiestoTransporte : ManifiestoTransporte, edicionAvanzada : boolean) : void
+listarReservasAsignadasTransportista(idTransportista : int, semana : Date) : List<Reserva>
+listarReservasDia(dia : int, semana : Date) : List<Reserva>
+listarReservasFiltro(idTipoFaena : Integer, estadoReserva : EstadoReserva, dia : int, semana : Date) : List<Reserva>
+calcularEstadoReserva(estadoVisacion : boolean, estadoTransporte : boolean) : EstadoReserva
+calcularEstadoReserva(idReserva : int) : EstadoReserva
+actualizarEstadoReserva(idReserva : int) : void
+init7() : void
-insertarDetalleReserva(reserva : Reserva, idReserva : Integer) : void
-insertarReserva(reserva : Reserva) : Reserva
-obtenerCodigoReserva(reserva : Reserva) : String
+registrarReserva2(reserva : Reserva) : Reserva
-poseeInformacionAdicional(reserva : Reserva) : boolean
-validarBeanReserva(reserva : Reserva) : void
-validarReserva(reserva : Reserva) : Set<ConstraintViolation<Reserva>>
+recuperarReserva2(idReserva : int) : Reserva
+aprobarReserva2(idReserva : int) : void
+cancelarReserva2(idReserva : int) : void
+listarReservas2(semana : Date, idReservador : Integer) : List<Reserva>
+listarReservas2(semana : Date) : List<Reserva>
+listarReservasAsociadasFaena2(idFaena : int) : List<Reserva>
+modificarHorarioReserva2(reserva : Reserva, dia : Integer, zonaHoraria : Integer) : void
+registrarManifiestoTransporte2(manifiestoTransporte : ManifiestoTransporte) : void
-validarBeanManifiestoTransporte(manifiestoTransporte : ManifiestoTransporte) : void
-validarManifiestoTransporte(manifiestoTransporte : ManifiestoTransporte) : Set<ConstraintViolation<ManifiestoTransporte...
+recuperarManifiestoTransporte2(idReserva : int) : ManifiestoTransporte
+modificarManifiestoTransporte2(manifiestoTransporte : ManifiestoTransporte, edicionAvanzada : boolean) : void
+listarReservasAsignadasTransportista2(idTransportista : int, semana : Date) : List<Reserva>
+listarReservasDia2(dia : int, semana : Date) : List<Reserva>
+listarReservasFiltro2(idTipoFaena : Integer, estadoReserva : EstadoReserva, dia : int, semana : Date) : List<Reserva>
+calcularEstadoReserva2(estadoVisacion : boolean, estadoTransporte : boolean) : EstadoReserva
+calcularEstadoReserva2(idReserva : int) : EstadoReserva
-existeReserva(idReserva : int) : boolean
-tieneTransporteEspecificado(idReserva : int) : boolean
-obtenerDocumentoAduaneroReserva(idReserva : int) : String
-obtenerCodigoContenedorReserva(idReserva : int) : String
+actualizarEstadoReserva2(idReserva : int) : void

```

```
+recuperarContenedor(codigo : String) : Contenedor
+validarContenedor(codigo : String) : boolean
+validarDocumentoAduanero(codigoContenedor : String, documentoAduanero : String) : boolean
+validarVisacionContenedor(codigoContenedor : String) : boolean
+validarEjecucionContenedor(codigoContenedor : String) : boolean
+recuperarContenedor2(codigo : String) : Contenedor
+validarContenedor2(codigo : String) : boolean
+validarDocumentoAduanero2(codigoContenedor : String, documentoAduanero : String) : boolean
+validarVisacionContenedor2(codigoContenedor : String) : boolean
+validarEjecucionContenedor2(codigoContenedor : String) : boolean
```