

Trabajo práctico integral UTN MDP 2017
Metodología de sistemas

Sistema de pedidos de cerveza artesanal web

Integrantes:

Garguir Sergio

Soria Rodrigo

Vega Gastón

Ámbito del sistema

El nombre del sistema a desarrollar es ChoppenHauer.

El sistema es principalmente un sitio web de E-commerce orientado a empresas de cerveza artesanal en el que los clientes (usuarios registrados) poseen un carrito de compras en el que pueden almacenar productos para luego adquirirlos directamente mediante la web, definiendo un método de envío a una dirección en específico o retirándolo directamente en la sucursal.

El sistema contará también con un backend personalizado exclusivo para empleados en el que podrán gestionar desde la ubicación y el contacto de las sucursales hasta los más finos detalles respecto a las cervezas y envases; su precio por litro, en qué medida afecta el tipo de envase al precio del producto final, qué envases estarán disponibles para qué cervezas y dar de alta, modificar o eliminar las ya existentes.

También los administradores serán capaces de gestionar los tipos de permisos para determinar una jerarquía entre las cuentas, conocidos como roles. Y dar de alta a otros empleados como también a otros clientes. Incluso se espera que puedan ver, eliminar, y/o modificar sus datos, siempre con las debidas precauciones, por si existe algún inconveniente con algún usuario.

El sistema no se encargará de la facturación del producto, sino más bien de la comunicación entre el usuario y la empresa propietaria, el cliente podrá ver y adquirir productos, incluso comprarlos sistemáticamente, pero el sistema no se encarga de emitir órdenes de factura ni ticket alguno y no está en los planes diseñarlo por el momento para ello.

Objetivos del sistema

Se espera, que mediante una interfaz sencilla e intuitiva, cualquier tipo de usuario sea capaz de entender y comprender el funcionamiento del sistema y sepa guiarse a través del menú y los botones de redirección. Todo esto con el objetivo de que el sistema sea capaz de atraer a la mayor cantidad de público posible para la empresa propietaria e incentivar a los mismos a adquirir los productos que en él se expongan. Si bien el objetivo principal es atraer a todo tipo de posible consumidor, el sistema está, debido al contenido y finalidad del mismo, orientado a un público mayor de edad (18 años, según la ley en la República Argentina, país de origen).

Como objetivo secundario, gracias a la finalidad que se espera dar por el lado del backend, se pretende que el sistema también funcione eficientemente como un sistema de gestión y administración de los productos en venta y los empleados del mismo, permitiendo definir qué se vende, qué no, cómo se vende, qué sucursales están disponibles y quiénes serán sus usuarios.

Suposiciones y dependencias

Puede ocurrir que algunos aspectos referentes al sistema varíen, como la plataforma en la que se visualizan las páginas. Si bien está destinado a exploradores y la visión principal es su visibilidad a través de un ordenador, no se descarta, y se estima, su desarrollo con tecnologías Responsive, para su cómoda visualización en cualquier tipo de pantallas.

Se ha decidido, para solventar esto, usar y depender de librerías bootstrap, que facilitan la aplicación de estas tecnologías.

Se estiman pocas probabilidades de que pueda variar algún que otro detalle interno, como el acceso a la base de datos, o la forma en que se calculan los precios de cada producto en específico o en conjunto (carrito). Pero debido a la arquitectura del mismo se prevé que cualquier cambio en las leyes de negocio o persistencia de datos pueda ser fácilmente adaptado.

Table of Contents

Functional Requirements	2
Functional Requirements diagram	2
RF_001 Alta y modificación de cervezas	2
RF_002 Eliminado de cervezas	3
RF_003 Alta y modificacion de envases	3
RF_004 Eliminado de Envases	3
RF_005 Gestión de clientes	4
RF_006 Gestión de roles	4
RF_007 Gestión de sucursales	5
RF_008 Geolocalización de las sucursales.	5
RF_009 Alta de cliente pública	6
RF_010 Carrito de compras "Pedido"	6
RF_011 Calculo del valor de los productos	6
RF_012 agregar producto al carrito	7
RF_013 Calculo del precio final del pedido	7
RF_014 Envío	7
RF_015 Compra	8
RF_016 Verificación de pedido	8
RF_017 Baja de cliente pública	9
RF_018 Clasificación de pedidos	9
RF_019 Método de envío	9
RF_020 Lista de pedidos	10
RF_021 Eliminación de roles	10
RF_022 Listado de cervezas	11
RF_023 Listado de envases	11
RF_024 Listado de roles	11
RF_025 Listado de sucursales	12
RF_026 Listado de usuarios y personal	12
RF_027 Estado de cervezas y envases.	12
RF_028 Envases por Cerveza	13
RF_029 Lista de cervezas para la venta	13
RF_030 Página de inicio.	13
RF_031 consultar pedido	14
RF_032 Ver perfil Cliente	14
RF_033 Gestión de personal	15
RF_034 ver cantidad de litros vendidos por cerveza entre fechas	15

Functional Requirements

Package in package 'Requirements Model'

Functional Requirements
Version 1.0 Phase 1.0 Proposed
usuario created on 11/10/2017. Last modified 11/10/2017

Functional Requirements diagram

Custom diagram in package 'Functional Requirements'

Functional Requirements
Version 1.0
usuario created on 11/10/2017. Last modified 23/11/2017

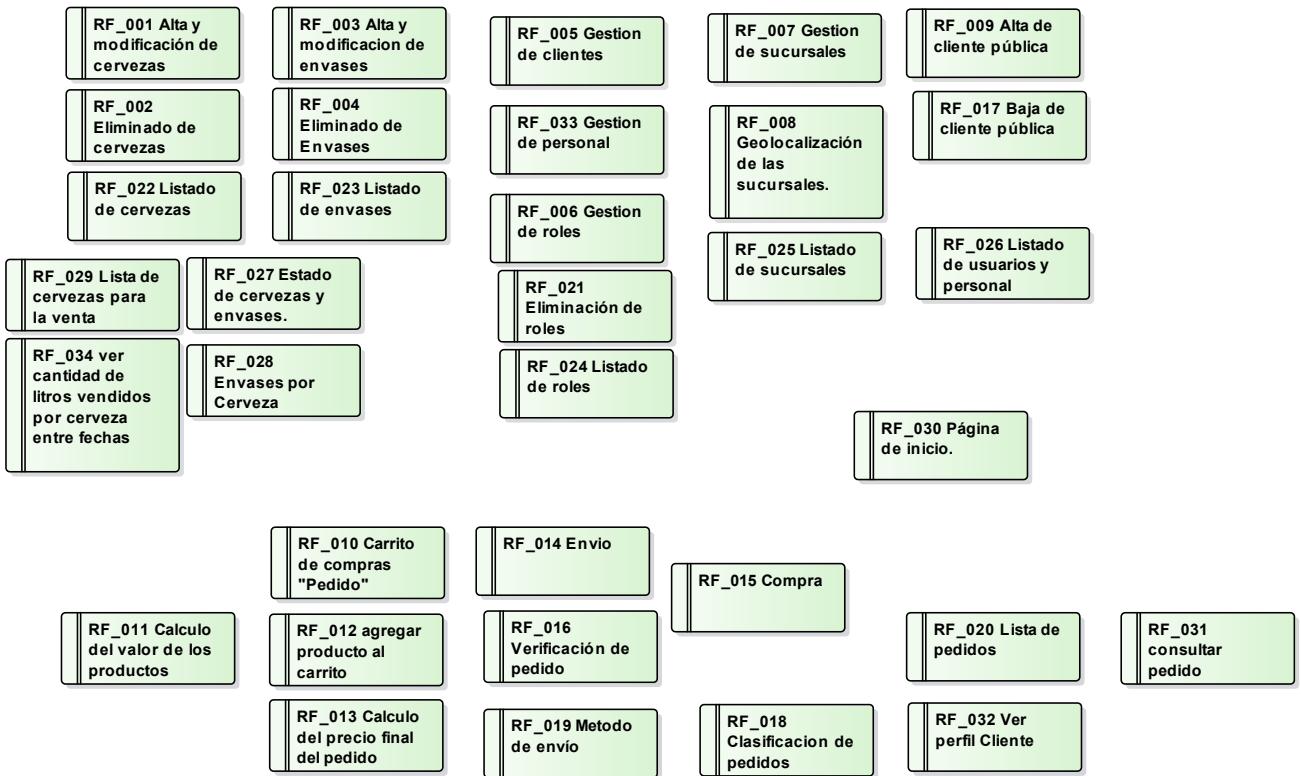


Figure 1: Functional Requirements

RF_001 Alta y modificación de cervezas

Requirement «Functional» in package 'Functional Requirements'

El **administrador** se encargará del alta de los tipos de cerveza, indicando su nombre, descripción, una imagen y el precio por litro de la misma. Podrá también modificar un tipo de cerveza de la misma manera.

RF_001 Alta y modificación de cervezas
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 20/11/2017
Alias Alta y modificación de cervezas
Keywords RF_001

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_002 Alta de cervezas to «Functional» RF_001 Alta y modificación de cervezas [Direction is 'Source -> Destination'.]
⇒ Realization from CU_013 Modificar cerveza to «Functional» RF_001 Alta y modificación de cervezas [Direction is 'Source -> Destination'.]

RF_002 Eliminado de cervezas

Requirement «Functional» in package 'Functional Requirements'

El **administrador** podrá borrar un tipo de cerveza.

RF_002 Eliminado de cervezas
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 24/10/2017
Alias Eliminado de cervezas
Keywords RF_002

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_012 Ver listado de cervezas to «Functional» RF_002 Eliminado de cervezas [Direction is 'Source -> Destination'.]

RF_003 Alta y modificacion de envases

Requirement «Functional» in package 'Functional Requirements'

El administrador se encargará del alta de los envases , indicando su nombre, capacidad, imagen y el coeficiente del mismo. Podrá también modificar un envase de la misma manera.

RF_003 Alta y modificacion de envases
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 20/11/2017
Alias Alta y modificacion de envases
Keywords RF_003

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_020 Modificar envase to «Functional» RF_003 Alta y modificacion de envases [Direction is 'Source -> Destination'.]
⇒ Realization from CU_007 Alta de envases to «Functional» RF_003 Alta y modificacion de envases [Direction is 'Source -> Destination'.]

RF_004 Eliminado de Envases

Requirement «Functional» in package 'Functional Requirements'

El **administrador** podrá eliminar un envase.

RF_004 Eliminado de Envases
Versión 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 24/10/2017
Alias Eliminado de Envases
Keywords RF_004

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_018 Ver listado de envases to «Functional» RF_004 Eliminado de Envases
 - [Direction is 'Source -> Destination'.]

RF_005 Gestión de clientes

Requirement «Functional» in package 'Functional Requirements'

El **administrador** se puede encargar del alta de clientes. Podrá también modificar un usuario de ser necesario. Además podrá eliminar un usuario ya registrado pero sólo si este no tiene pedidos que aún no hayan sido finalizados/cancelados. (*Ver RF_020*).

RF_005 Gestión de clientes
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 20/11/2017
Alias Gestión de clientes
Keywords RF_005

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_003 Alta de clientes to «Functional» RF_005 Gestión de clientes
 - [Direction is 'Source -> Destination'.]
- ⇒ Realization from CU_023 Ver listado de clientes to «Functional» RF_005 Gestión de clientes
 - [Direction is 'Source -> Destination'.]
- ⇒ Realization from CU_004 Modificación de Clientes to «Functional» RF_005 Gestión de clientes
 - [Direction is 'Source -> Destination'.]

RF_006 Gestión de roles

Requirement «Functional» in package 'Functional Requirements'

El **administrador** puede dar de alta un rol con su respectiva descripción, modificarlo o eliminar uno. Para borrar un rol (*Ver RF_021*).

RF_006 Gestión de roles
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 14/10/2017
Alias Gestión de roles
Keywords RF_006

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_015 Eliminar un rol to «Functional» RF_006 Gestión de roles [Direction is 'Source -> Destination'.]
⇒ Realization from CU_001 Alta de roles to «Functional» RF_006 Gestión de roles [Direction is 'Source -> Destination'.]
⇒ Realization from CU_016 Modificar un rol to «Functional» RF_006 Gestión de roles [Direction is 'Source -> Destination'.]

RF_007 Gestión de sucursales

Requirement «Functional» in package 'Functional Requirements'

El **administrador** deberá encargarse del alta de las sucursales, indicando su ubicación, localidad, nombre y un teléfono de contacto. Podrá modificar los datos de las demás sucursales y también dar de baja una sucursal de ser necesario.

RF_007 Gestión de sucursales
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 20/11/2017
Alias Gestión de sucursales
Keywords RF_007

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_019 Listado de sucursales to «Functional» RF_007 Gestión de sucursales [Direction is 'Source -> Destination'.]
⇒ Realization from CU_014 Modificar sucursal to «Functional» RF_007 Gestión de sucursales [Direction is 'Source -> Destination'.]
⇒ Realization from CU_008 Alta de sucursales. to «Functional» RF_007 Gestión de sucursales [Direction is 'Source -> Destination'.]

RF_008 Geolocalización de las sucursales.

Requirement «Functional» in package 'Functional Requirements'

Se espera poder acceder a un mapa de las sucursales por medio de **Google Maps**

RF_008 Geolocalización de las sucursales.
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 20/11/2017
Alias Geolocalización de las sucursales.
Keywords RF_008

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_014 Modificar sucursal to «Functional» RF_008 Geolocalización de las sucursales. [Direction is 'Source -> Destination'.]
⇒ Realization from CU_008 Alta de sucursales. to «Functional» RF_008 Geolocalización de las sucursales. [Direction is 'Source -> Destination'.]

RF_009 Alta de cliente pública

Requirement «Functional» in package 'Functional Requirements'

Cualquier persona puede registrarse en el sistema como un cliente, de manera publica directamente desde el navegador.

RF_009 Alta de cliente pública
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 12/10/2017
Alias Alta de cliente pública
Keywords RF_009

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_003 Alta de clientes to «Functional» RF_009 Alta de cliente pública [Direction is 'Source -> Destination'.]

RF_010 Carrito de compras "Pedido"

Requirement «Functional» in package 'Functional Requirements'

El Cliente tendrá un carrito de compras llamado "Pedido". El cual tendrá todos los productos que el cliente haya solicitado.
almacenando cada producto como una "línea de pedido" (**ver RF_012**).

RF_010 Carrito de compras "Pedido"
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 20/11/2017
Alias Carrito de compras "Pedido"
Keywords RF_010

INCOMING STRUCTURAL RELATIONSHIPS
⇒ Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_010 Carrito de compras "Pedido" [Direction is 'Source -> Destination'.]

RF_011 Calculo del valor de los productos

Requirement «Functional» in package 'Functional Requirements'

Se deberá calcular el valor de un producto (**envase+cerveza * cant**) De manera automática de acuerdo al precio por litro de la cerveza y un coeficiente del envase. Éste ultimo cumple con la función de aumentar o disminuir el precio final del producto.

RF_011 Calculo del valor de los productos
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 18/10/2017
Alias Calculo del valor de los productos
Keywords RF_011

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_009 Agregar cerveza al carrito. to «Functional» RF_011 Calculo del valor de los productos
[Direction is 'Source -> Destination'.]

RF_012 agregar producto al carrito

Requirement «Functional» in package 'Functional Requirements'

Un cliente deberá ser capaz de seleccionar un tipo de cerveza, con un envase a elección y la cantidad que desee para agregar a su carrito. Creando así varias líneas de pedidos.

Se considera linea de pedido a: **Cerveza + Envase + Cantidad + subtotal**.

RF_012 agregar producto al carrito
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 12/10/2017
Alias agregar producto al carrito
Keywords RF_012

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_009 Agregar cerveza al carrito. to «Functional» RF_012 agregar producto al carrito
[Direction is 'Source -> Destination'.]

RF_013 Calculo del precio final del pedido

Requirement «Functional» in package 'Functional Requirements'

Los pedidos deberán tener el precio final de la suma de todos los productos que el cliente haya seleccionado (**ver RF_012**). Y si se desea (opcional) alguna especie de descuento.

RF_013 Calculo del precio final del pedido
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 20/11/2017
Alias Calculo del precio final del pedido
Keywords RF_013

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_009 Agregar cerveza al carrito. to «Functional» RF_013 Calculo del precio final del pedido
[Direction is 'Source -> Destination'.]

RF_014 Envio

Requirement «Functional» in package 'Functional Requirements'

Un pedido puede tener un envío específico a una determinada ubicación. Por lo que la implementación de los mismos es necesaria. Si el usuario no especifica la información, se tomará como destino por defecto la dirección del cliente mismo (A menos que sea retiro en sucursal) (*ver RF_019*) .

RF_014 Envio
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 20/11/2017
Alias Envio
Keywords RF_014

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_014 Envio

[Direction is 'Source -> Destination'.]

RF_015 Compra

Requirement «Functional» in package 'Functional Requirements'

El cliente, una vez esté satisfecho con su pedido (*ver RF_016*) podrá realizar la operación de compra. Y se creará un registro con el pedido ya realizado, o como un pedido en espera de envío. (*ver RF_019 y RF_018*).

RF_015 Compra
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 18/10/2017
Alias Compra
Keywords RF_015

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_015 Compra

[Direction is 'Source -> Destination'.]

RF_016 Verificación de pedido

Requirement «Functional» in package 'Functional Requirements'

El cliente deberá poder revisar su pedido antes de apretar el botón para que se proceda la compra, dándole la oportunidad de cancelar líneas de pedido que haya creado.

El cliente deberá seleccionar su método de envío preferido (*ver RF_019*)..

RF_016 Verificación de pedido
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 20/11/2017
Alias Verificación de pedido
Keywords RF_016

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_016 Verificación de pedido

[Direction is 'Source -> Destination'.]

RF_017 Baja de cliente pública

Requirement «Functional» in package 'Functional Requirements'

El cliente puede darse de baja en cualquier momento directamente desde el sitio web. Pero siempre y cuando no tenga pedidos pendientes sin pagar. (*Ver RF_020*)

RF_017 Baja de cliente pública
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 14/10/2017
Alias Baja de cliente pública
Keywords RF_017

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_005 Darse de baja to «Functional» RF_017 Baja de cliente pública

RF_018 Clasificacion de pedidos

Requirement «Functional» in package 'Functional Requirements'

el cliente tendrá pedidos, y los pedidos tienen un estado, el cual puede ser o solicitado, o en proceso, o enviado o finalizado.

RF_018 Clasificacion de pedidos
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 23/11/2017
Alias Clasificacion de pedidos
Keywords RF_018

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_029 consultar pedido to «Functional» RF_018 Clasificacion de pedidos
- ⇒ Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_018 Clasificacion de pedidos
- ⇒ Realization from CU_028 Ver historial de pedidos to «Functional» RF_018 Clasificacion de pedidos

RF_019 Metodo de envío

Requirement «Functional» in package 'Functional Requirements'

El cliente deberá poder optar entre realizar un envío a una determinada **ubicación específica**, elegir si **recibir el envío en su domicilio**, o realizar una **recarga por sucursal**.

Para ésta última tendrá que seleccionar la sucursal deseada desde **google maps** o un menú desplegable

RF_019 Metodo de envío
Version 1.0 Phase 1.0 Proposed

Rodrigo created on 12/10/2017. Last modified 20/11/2017

Alias Metodo de envío

Keywords RF_019

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_019 Metodo de envío
 [Direction is 'Source -> Destination'.]

RF_020 Lista de pedidos*Requirement «Functional» in package 'Functional Requirements'*

Un cliente tendrá una lista de pedidos, la cuál prodrá examinar y ver cada pedido con su correspondiente estado.

RF_020 Lista de pedidos

Version 1.0 Phase 1.0 Proposed

Rodrigo Soria created on 14/10/2017. Last modified 23/11/2017

Alias Lista de pedidos

Keywords RF_020

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_023 Ver listado de clientes to «Functional» RF_020 Lista de pedidos
 [Direction is 'Source -> Destination'.]
- ⇒ Realization from CU_005 Darse de baja to «Functional» RF_020 Lista de pedidos
 [Direction is 'Source -> Destination'.]
- ⇒ Realization from CU_028 Ver historial de pedidos to «Functional» RF_020 Lista de pedidos
 [Direction is 'Source -> Destination'.]

RF_021 Eliminación de roles*Requirement «Functional» in package 'Functional Requirements'*

para poder eliminar un rol primero no deberá haber ninguna cuenta con el mismo. Se tendrá que asignar un rol diferente a aquellos que posean rol antes de proceder con la eliminación

RF_021 Eliminación de roles

Version 1.0 Phase 1.0 Proposed

Rodrigo Soria created on 14/10/2017. Last modified 14/10/2017

Alias Eliminación de roles

Keywords RF_021

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_015 Eliminar un rol to «Functional» RF_021 Eliminación de roles
 [Direction is 'Source -> Destination'.]

RF_022 Listado de cervezas

Requirement «Functional» in package 'Functional Requirements'

El administrador deberá ser capaz de ver un listado con los tipos de cerveza de la base de datos así como diversos botones para modificar, eliminar y cambiar el estado de las mismas. (ver RF_027)

RF_022 Listado de cervezas
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 20/11/2017
Alias Listado de cervezas
Keywords RF_022

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_012 Ver listado de cervezas to «Functional» RF_022 Listado de cervezas
[Direction is 'Source -> Destination'.]

RF_023 Listado de envases

Requirement «Functional» in package 'Functional Requirements'

El administrador deberá ser capaz de ver un listado con los envases de la base de datos así como diversos botones para modificar, eliminar y cambiar el estado de los mismos. (ver RF_027)

RF_023 Listado de envases
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 20/11/2017
Alias Listado de cervezas
Keywords RF_023

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_018 Ver listado de envases to «Functional» RF_023 Listado de envases
[Direction is 'Source -> Destination'.]

RF_024 Listado de roles

Requirement «Functional» in package 'Functional Requirements'

El administrador podrá ver una lista de los roles disponibles en la base de datos y botones para modificar o eliminar a los mismos.

RF_024 Listado de roles
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 19/10/2017
Alias Listado de roles
Keywords RF_024

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_017 Ver listado de roles to «Functional» RF_024 Listado de roles
[Direction is 'Source -> Destination'.]

RF_025 Listado de sucursales

Requirement «Functional» in package 'Functional Requirements'

El administrador deberá poder visualizar un listado con las sucursales almacenadas en el sistema, así como botones para modificarlas o eliminarlas.

RF_025 Listado de sucursales
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 19/10/2017
Alias Listado de sucursales
Keywords RF_025

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_019 Listado de sucursales to «Functional» RF_025 Listado de sucursales
[Direction is 'Source -> Destination'.]

RF_026 Listado de usuarios y personal

Requirement «Functional» in package 'Functional Requirements'

Un administrador deberá ser capaz de acceder a una lista con todas las cuentas del sistema, eligiendo si desea ver al personal o a los clientes. Se desea que pueda acceder a la información del mismo. Así como también poder borrar o modificar la información de estos.

RF_026 Listado de usuarios y personal
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 20/11/2017
Alias Listado de usuarios y personal
Keywords RF_026

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_021 Ver listado del personal to «Functional» RF_026 Listado de usuarios y personal
[Direction is 'Source -> Destination'.]
- ⇒ Realization from CU_023 Ver listado de clientes to «Functional» RF_026 Listado de usuarios y personal
[Direction is 'Source -> Destination'.]

RF_027 Estado de cervezas y envases.

Requirement «Functional» in package 'Functional Requirements'

tanto la cerveza como los envases, al ser productos para la venta, deberán tener un estado; es decir, si están habilitados o deshabilitados para la venta.

RF_027 Estado de cervezas y envases.
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 20/10/2017
Alias Estado de cervezas y envases
Keywords RF_027

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_018 Ver listado de envases to «Functional» RF_027 Estado de cervezas y envases.
[Direction is 'Source -> Destination'.]

- ⇒ Realization from CU_012 Ver listado de cervezas to «Functional» RF_027 Estado de cervezas y envases.
[Direction is 'Source -> Destination'.]

RF_028 Envases por Cerveza

Requirement «Functional» in package 'Functional Requirements'

Cada cerveza se ha de vender en determinados envases específicos, y no en cualquiera. Debido a que lógicamente, no se puede vender cualquier cerveza en cualquier envase.

RF_028 Envases por Cerveza
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 20/11/2017. Last modified 20/11/2017
Alias Envases por Cerveza
Keywords RF_028

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_024 Administrar envases por cervezas to «Functional» RF_028 Envases por Cerveza
[Direction is 'Source -> Destination'.]

RF_029 Lista de cervezas para la venta

Requirement «Functional» in package 'Functional Requirements'

Cualquier usuario, registrado o no, deberá ser capaz de acceder a una lista con las cervezas disponibles para la venta

RF_029 Lista de cervezas para la venta
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias Lista de cervezas para la venta
Keywords RF_029

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from CU_026 Ver productos to «Functional» RF_029 Lista de cervezas para la venta
[Direction is 'Source -> Destination'.]

RF_030 Página de inicio.

Requirement «Functional» in package 'Functional Requirements'

Debe de existir una página de inicio general, igual para todos los tipos de usuarios, sean empleados o no, o estén conectados o no, que prevea acceso a la mayoría de las funciones que un usuario pueda hacer.

A su vez, debe existir otra segunda página de inicio pero del lado del backend para cuando un administrador decida entrar.

RF_030 Página de inicio.
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias Página de inicio
Keywords RF_030

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Realization from CU_025 Ir a inicio to «Functional» RF_030 Página de inicio.

[Direction is 'Source -> Destination'.]

RF_031 consultar pedido

Requirement «Functional» in package 'Functional Requirements'

El administrador puede consultar un pedido de un cliente y modificar su estado.

A su vez, un cliente puede consultar un pedido para ver qué es lo que en él compró y su estado, pero no puede realizar modificaciones.

RF_031 consultar pedido
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias consultar pedido
Keywords RF_031

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Realization from CU_029 consultar pedido to «Functional» RF_031 consultar pedido

[Direction is 'Source -> Destination'.]

RF_032 Ver perfil Cliente

Requirement «Functional» in package 'Functional Requirements'

El usuario deberá ser capaz de ver su perfil con la información de su cuenta.

también debería ser capaz de acceder a una lista con los pedidos que haya realizado. (historial). (**ver RF_020**).

RF_032 Ver perfil Cliente
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 18/10/2017. Last modified 23/11/2017
Alias Ver perfil Cliente
Keywords RF_032

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Realization from CU_011 Ver perfil to «Functional» RF_032 Ver perfil Cliente

[Direction is 'Source -> Destination'.]

RF_033 Gestión de personal

Requirement «Functional» in package 'Functional Requirements'

El administrador podrá dar de alta a nuevos miembros del staff. Así como modificarlos y darlos de baja.

RF_033 Gestión de personal
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 11/10/2017. Last modified 23/11/2017
Alias Gestión de personal
Keywords RF_033

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Realization from CU_021 Ver listado del personal to «Functional» RF_033 Gestión de personal
[Direction is 'Source -> Destination'.]

⇒ Realization from CU_022 Alta de personal to «Functional» RF_033 Gestión de personal
[Direction is 'Source -> Destination'.]

⇒ Realization from CU_006 Modificación de Personal to «Functional» RF_033 Gestión de personal
[Direction is 'Source -> Destination'.]

RF_034 ver cantidad de litros vendidos por cerveza entre fechas

Requirement «Functional» in package 'Functional Requirements'

El administrador deberá ser capaz de ver cuantos litros se vendieron de determinada cerveza entre dos fechas específicas

RF_034 ver cantidad de litros vendidos por cerveza entre fechas
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias ver cantidad de litros vendidos por cerveza
Keywords RF_034

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Realization from CU_030 ver cantidad de litros vendidos to «Functional» RF_034 ver cantidad de litros vendidos por cerveza entre fechas
[Direction is 'Source -> Destination'.]

Table of Contents

Non-Functional Requirements	2
Non-Functional Requirements diagram	2
RNF_001 Persistencia de datos	2
RNF_002 Tecnologías para la GUI	2
RNF_003 Responsive GUI	3
RNF_004 Arquitectura en capas	3

Non-Functional Requirements

Package in package 'Requirements Model'

Non-Functional Requirements
Version 1.0 Phase 1.0 Proposed
usuario created on 11/10/2017. Last modified 11/10/2017

Non-Functional Requirements diagram

Custom diagram in package 'Non-Functional Requirements'

Non-Functional Requirements
Version 1.0
Rodrigo created on 11/10/2017. Last modified 23/11/2017



Figure 1: Non-Functional Requirements

RNF_001 Persistencia de datos

Requirement «Functional» in package 'Non-Functional Requirements'

La persistencia de datos deberá ser en una base de datos de MySql, probando antes el funcionamiento utilizando listas

RNF_001 Persistencia de datos
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 12/10/2017
Alias Persistencia de datos
Keywords RNF_001

RNF_002 Tecnologías para la GUI

Requirement «Functional» in package 'Non-Functional Requirements'

Se deberá decidir que tipo de tecnologías se utilizarán para crear la interfaz de usuario gráfica.

RNF_002 Tecnologías para la GUI
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 24/10/2017
Alias Tecnologías para la GUI
Keywords RNF_002

RNF_003 Responsive GUI

Requirement «Functional» in package 'Non-Functional Requirements'

En base a la tecnología para la GUI (*ver RNF_002*) se codificarán las vistas de manera que la interfaz sea visualizada correctamente en dispositivos móviles como en una pc o tablet.

RNF_003 Responsive GUI
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 12/10/2017. Last modified 12/10/2017
Alias Responsive GUI
Keywords RNF_003

RNF_004 Arquitectura en capas

Requirement «Functional» in package 'Non-Functional Requirements'

Programación en capas de la aplicación respetando la arquitectura de 3 capas lógicas vista durante la cursada. Esto implica el desarrollo de las clases que representen las entidades del modelo, controladoras de los casos de uso, las vistas y la capa de acceso a datos.

RNF_004 Arquitectura en capas
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias Arquitectura en capas
Keywords RNF_004

Arquitectura del sistema

El sistema está diseñado en un entorno web, con estructura en capas lógicas y siguiendo los principios GRASP, usando patrones como Controlador, Fabricación pura y Experto en información. Debido a la estructura en capas lógicas (presentación, reglas de negocio, persistencia de datos), la adaptación a los cambios será mucho más rápida y flexible al tener todo separado, los controladores se encargan de las reglas de negocio y actúan como mediadores entre la interfaz gráfica de usuario, que es la capa de presentación, y la capa de persistencia, que contiene las clases que se encargan de manipular los objetos en la base de datos.

Table of Contents

Use Case Model	2
Use Case Model diagram	2
Administrador	2
Cliente	4
Usuario no registrado	5
CU_001 Alta de roles	5
CU_002 Alta de cervezas	7
CU_003 Alta de clientes	8
CU_004 Modificación de Clientes	9
CU_005 Darse de baja	10
CU_006 Modificación de Personal	11
CU_007 Alta de envases	12
CU_008 Alta de sucursales	14
CU_009 Agregar cerveza al carrito.	15
CU_010 Ver carrito y confirmar compra	16
CU_011 Ver perfil	19
CU_012 Ver listado de cervezas	20
CU_013 Modificar cerveza	21
CU_014 Modificar sucursal	22
CU_015 Eliminar un rol	23
CU_016 Modificar un rol	25
CU_017 Ver listado de roles	25
CU_018 Ver listado de envases	27
CU_019 Listado de sucursales	28
CU_020 Modificar envase	29
CU_021 Ver listado del personal	30
CU_022 Alta de personal	32
CU_023 Ver listado de clientes	33
CU_024 Administrar envases por cervezas	34
CU_025 Ir a inicio	35
CU_026 Ver productos	37
CU_027 Iniciar sesión	37
CU_028 Ver historial de pedidos	38
CU_029 consultar pedido	39
CU_030 ver cantidad de litros vendidos	41

Use Case Model

Package in package 'Model'

Use Case Model
Version Phase 1.0 Proposed
usuario created on 11/10/2017. Last modified 11/10/2017

Use Case Model diagram

Use Case diagram in package 'Use Case Model'

Use Case Model
Version 1.0
usuario created on 11/10/2017. Last modified 23/11/2017

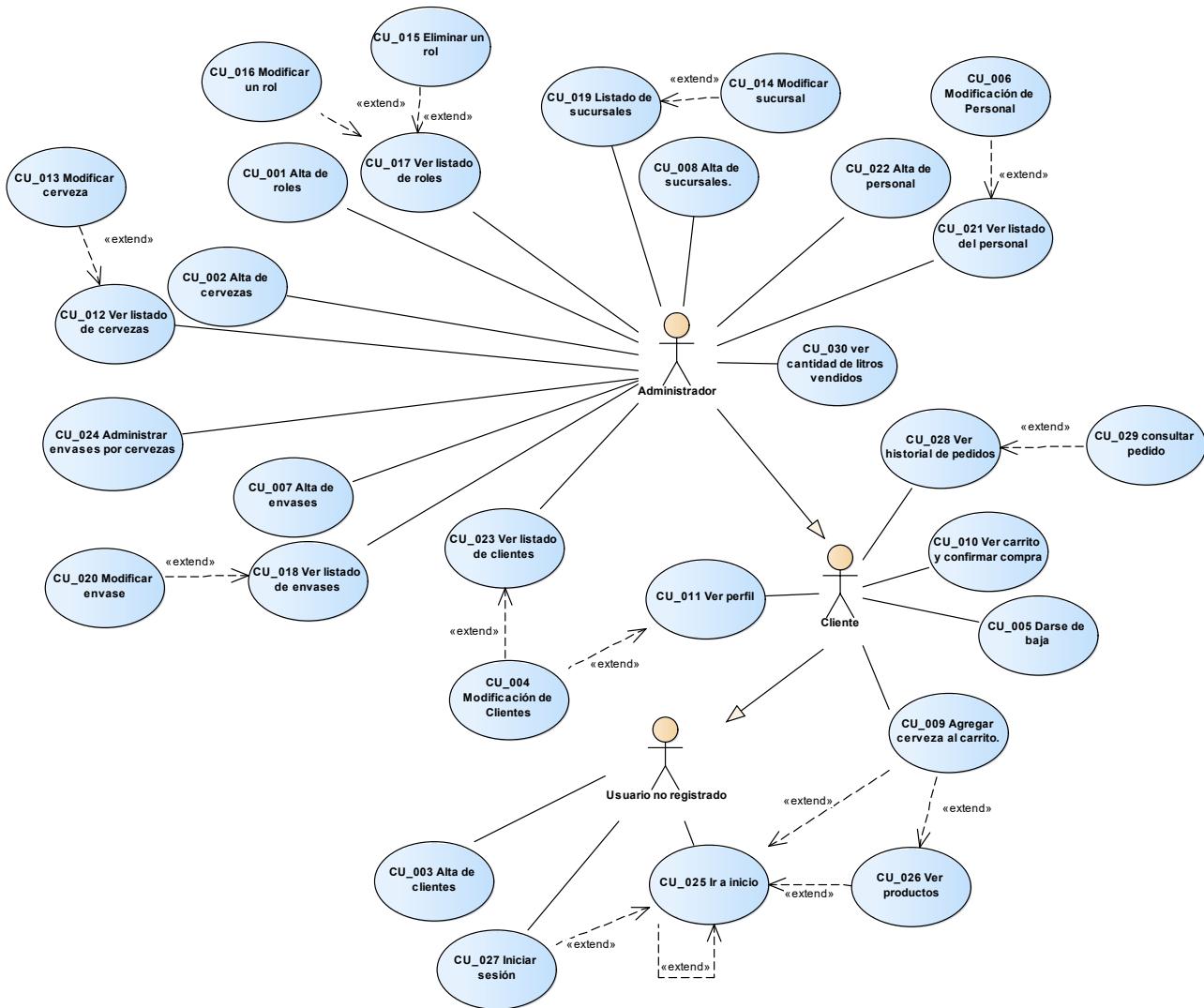


Figure 1: Use Case Model

Administrador

Actor in package 'Use Case Model'

El administrador del sistema, quien se encarga de la gestión general del software.

Administrador
Version 1.0 Phase 1.0 Proposed
usuario created on 11/10/2017. Last modified 11/10/2017

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from Administrador to Cliente

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_023 Ver listado de clientes : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_022 Alta de personal : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_002 Alta de cervezas : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_024 Administrar envases por cervezas : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_008 Alta de sucursales. : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_018 Ver listado de envases : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_030 ver cantidad de litros vendidos : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_021 Ver listado del personal : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_012 Ver listado de cervezas : UseCase, Public

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

CONNECTORS	
To:	CU_019 Listado de sucursales : UseCase, Public
 UseCaseLink	Source -> Destination From: Administrador : Actor, Public To: CU_001 Alta de roles : UseCase, Public
 UseCaseLink	Source -> Destination From: Administrador : Actor, Public To: CU_007 Alta de envases : UseCase, Public
 UseCaseLink	Source -> Destination From: Administrador : Actor, Public To: CU_017 Ver listado de roles : UseCase, Public

Cliente

Actor in package 'Use Case Model'

Un usuario registrado que puede realizar determinadas funciones y comprar en la página

Cliente
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 13/10/2017. Last modified 13/10/2017
Alias Cliente

OUTGOING STRUCTURAL RELATIONSHIPS	
 Generalization from Cliente to Usuario no registrado	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
 Generalization from Administrador to Cliente	[Direction is 'Source -> Destination'.]

CONNECTORS	
 UseCaseLink	Source -> Destination From: Cliente : Actor, Public To: CU_028 Ver historial de pedidos : UseCase, Public
 UseCaseLink	Source -> Destination From: Cliente : Actor, Public To: CU_009 Agregar cerveza al carrito. : UseCase, Public
 UseCaseLink	Source -> Destination From: Cliente : Actor, Public To: CU_005 Darse de baja : UseCase, Public

CONNECTORS	
 UseCaseLink	Source -> Destination From: Cliente : Actor, Public To: CU_011 Ver perfil : UseCase, Public
 UseCaseLink	Source -> Destination From: Cliente : Actor, Public To: CU_010 Ver carrito y confirmar compra : UseCase, Public

Usuario no registrado

Actor in package 'Use Case Model'

Un usuario no registrado, que puede navegar por la página, pero deberá de registrarse para acceder a las funciones del cliente

Usuario no registrado
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 13/10/2017. Last modified 23/11/2017
Alias Usuario

INCOMING STRUCTURAL RELATIONSHIPS	
 Generalization	from Cliente to Usuario no registrado [Direction is 'Source -> Destination'.]

CONNECTORS	
 UseCaseLink	Source -> Destination From: Usuario no registrado : Actor, Public To: CU_025 Ir a inicio : UseCase, Public
 UseCaseLink	Source -> Destination From: Usuario no registrado : Actor, Public To: CU_027 Iniciar sesión : UseCase, Public
 UseCaseLink	Source -> Destination From: Usuario no registrado : Actor, Public To: CU_003 Alta de clientes : UseCase, Public

CU_001 Alta de roles

UseCase in package 'Use Case Model'

El administrador podrá dar de alta un rol, definiendo el nombre del mismo, la descripción y el tipo de permisos.

CU_001 Alta de roles
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 13/10/2017. Last modified 20/11/2017
Alias Alta de roles

EXTERNAL REQUIREMENTS

Requirement. RF_006 Gestión de roles

El **administrador** puede dar de alta un rol con su respectiva descripción, modificarlo o eliminar uno.
Para borrar un rol (*Ver RF_021*).

[Stereotype is «Functional»]

SCENARIOS

Basic Path. Alta

El administrador dispondrá de una ventana con dos cuadros de texto que deberá llenar para luego apretar en un botón de "cargar al sistema" por el cuál se dará de alta el rol.

1. El usuario ingresa nombre, descripción de un rol, selecciona el tipo de permisos que posee, y apreta el botón de dar de alta.
Exception: 1a. Campo vacío
2. El sistema revisa si ya existe en la base de datos y si éste no existe mostrará un mensaje diciendo que el rol fue dado de alta correctamente.
Exception: 2a. Ya existe en la base de datos

Exception. Ya existe en la base de datos

1. El sistema muestra un mensaje de error, indicando que ya existe en la base de datos. con un botón de reintentar que lo regresará a la página de alta.

Exception. Campo vacío

1. Si un campo está vacío, el usuario se verá obligado a rellenarlo. El sistema marcará aquel dato restante en rojo y no le permitirá enviar los datos.

CONSTRAINTS

Pre-condition. Admin logueado.

El administrador deberá estar logueado para que pueda acceder.

[Approved, Weight is 0.]

OUTGOING STRUCTURAL RELATIONSHIPS

Realization from CU_001 Alta de roles to «Functional» RF_006 Gestión de roles

[Direction is 'Source -> Destination'.]

CONNECTORS

UseCaseLink Source -> Destination

From: Administrador : Actor, Public

To: CU_001 Alta de roles : UseCase, Public

CU_002 Alta de cervezas

UseCase in package 'Use Case Model'

El administrador podrá dar de alta un tipo de cerveza

CU_002 Alta de cervezas
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 13/10/2017. Last modified 19/10/2017
Alias Gestión de Cervezas
Keywords CU_002

EXTERNAL REQUIREMENTS

- Requirement. RF_001 Alta y modificación de cervezas

El **administrador** se encargará del alta de los tipos de cerveza, indicando su nombre, descripción, una imagen y el precio por litro de la misma. Podrá también modificar un tipo de cerveza de la misma manera.

[Stereotype is «Functional»]

SCENARIOS

Basic Path. Alta Cerveza

El administrador puede dar de alta una cerveza, indicando nombre, descripción, y precio por litro.

1. El usuario ingresa los datos y apreta en el botón de agregar cerveza.
2. El sistema verifica que la cerveza no exista en la base de datos (buscando por nombre). Y en caso de que no lo encuentre, lo cargará al sistema. Mostrará un mensaje de que fue agregado correctamente y un botón para agregar otro, que redireccionará a la página de alta.

Exception: 2a. La cerveza ya existe en el sistema

Exception: 2b. Hay campos vacíos o incorrectos

Exception. La cerveza ya existe en el sistema

1. El sistema mostrará un mensaje de error diciendo que la cerveza con ese nombre ya existe.

Exception. Hay campos vacíos o incorrectos

1. El sistema marca los cuadros correspondientes al problema y muestra un mensaje especificando que hay campos vacíos o incorrectos.

CONSTRAINTS

Pre-condition. El usuario está logueado como administrador.

El administrador deberá estar logueado para que pueda acceder.

[Approved, Weight is 0.]

OUTGOING STRUCTURAL RELATIONSHIPS

Realization from CU_002 Alta de cervezas to «Functional» RF_001 Alta y modificación de cervezas

[Direction is 'Source -> Destination'.]

CONNECTORS

 **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public
 To: CU_002 Alta de cervezas : UseCase, Public

CU_003 Alta de clientes

UseCase in package 'Use Case Model'

Tanto el administrador como cualquier usuario podrá registrarse en la página como un cliente. En el caso de administrador, éste da de alta una nueva cuenta.

CU_003 Alta de clientes
 Version 1.0 Phase 1.0 Proposed
 Rodrigo created on 13/10/2017. Last modified 20/11/2017
 Alias Alta de clientes
 Keywords CU_003

EXTERNAL REQUIREMENTS

Requirement. RF_005 Gestión de clientes

El **administrador** se puede encargar del alta de clientes. Podrá también modificar un usuario de ser necesario. Además podrá eliminar un usuario ya registrado pero sólo si este no tiene pedidos que aún no hayan sido finalizados/cancelados. (*Ver RF_020*).

[Stereotype is «Functional»]

Requirement. RF_009 Alta de cliente pública

Cualquier persona puede registrarse en el sistema como un cliente, de manera pública directamente desde el navegador.

[Stereotype is «Functional»]

SCENARIOS

 Basic Path. Alta de Cliente

El usuario podrá dar de alta una cuenta.

1. El usuario ingresa los datos correspondientes a su email, nombre, apellido, contraseña, dirección, localidad, teléfono y DNI y presiona en el botón de registrarse.
2. El sistema corroborará los datos, de estar todo bien, buscará en la base de datos si la cuenta, el nombre de usuario y el DNI existen. De no existir crea la cuenta correspondiente y muestra un mensaje de que el usuario ha sido registrado con éxito.
 Exception: 2a. La cuenta, o el DNI existen en el sistema

 Exception. La cuenta, o el DNI existen en el sistema

1. El sistema marcará los cuadros de texto correspondientes al DNI o la cuenta si ya existen según corresponda con los encontrados en la base de datos y mostrará de que ya existe en el sistema.

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS

➡ Realization from CU_003 Alta de clientes to «Functional» RF_005 Gestión de clientes
 [Direction is 'Source -> Destination'.]

➡ Realization from CU_003 Alta de clientes to «Functional» RF_009 Alta de cliente pública
 [Direction is 'Source -> Destination'.]

CONNECTORS

↗ UseCaseLink Source -> Destination
 From: Usuario no registrado : Actor, Public
 To: CU_003 Alta de clientes : UseCase, Public

CU_004 Modificación de Clientes

UseCase in package 'Use Case Model'

El usuario dispondrá de una vista como la del alta del cliente pero con la información correspondiente al usuario seleccionado y los campos habilitados para la edición.

CU_004 Modificación de Clientes
 Version 1.0 Phase 1.0 Proposed
 Rodrigo created on 13/10/2017. Last modified 20/11/2017
 Alias Modificación de Clientes
 Keywords CU_004

EXTERNAL REQUIREMENTS

Requirement. RF_005 Gestión de clientes

El **administrador** se puede encargar del alta de clientes. Podrá también modificar un usuario de ser necesario. Además podrá eliminar un usuario ya registrado pero sólo si este no tiene pedidos que aún no hayan sido finalizados/cancelados. (*Ver RF_020*).

[Stereotype is «Functional»]

SCENARIOS

HP Basic Path. Modificar Cliente

El administrador será capaz de modificar la información de un cliente. El objetivo es que lo haga cuando un cliente lo solicite.

1. El sistema carga una ventana como la del alta de cliente pero con la información del cliente seleccionado, los campos habilitados para su edición, y un botón para confirmar los cambios.
2. El usuario modifica algún (o ningún) campo y presiona en modificar
3. El sistema revisará que los datos estén correctamente ingresados y, si todo sale bien, modificará los datos correspondientes al cliente en la base de datos y mostrará un mensaje de que fue correctamente modificado.

Exception: 3a. Campos ingresados incorrectamente o vacíos

HP Exception. Campos ingresados incorrectamente o vacíos

1. El sistema marcará aquellos campos que hayan sido ingresados incorrectamente y mostrará un mensaje diciendo que hay campos incorrectos o vacíos.

CONSTRAINTS

Pre-condition. Administrador o usuario logueado

El administrador o el usuario deberá estar logueado para poder realizar las operaciones

[Approved, Weight is 0.]

OUTGOING STRUCTURAL RELATIONSHIPS

Realization from CU_004 Modificación de Clientes to «Functional» RF_005 Gestión de clientes

[Direction is 'Source -> Destination'.]

CONNECTORS

Extend «extend» Source -> Destination

From: CU_004 Modificación de Clientes : UseCase, Public
To: CU_023 Ver listado de clientes : UseCase, Public

Extend «extend» Source -> Destination

From: CU_004 Modificación de Clientes : UseCase, Public
To: CU_011 Ver perfil : UseCase, Public

CU_005 Darse de baja

UseCase in package 'Use Case Model'

Un cliente puede darse de baja. Siempre y cuando éste no tenga pedidos pendientes.

CU_005 Darse de baja
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 13/10/2017. Last modified 14/10/2017
Alias Darse de baja
Keywords CU_005

EXTERNAL REQUIREMENTS

Requirement. RF_017 Baja de cliente pública

El cliente puede darse de baja en cualquier momento directamente desde el sitio web. Pero siempre y cuando no tenga pedidos pendientes sin pagar. (*Ver RF_020*)

[Stereotype is «Functional»]

Requirement. RF_020 Lista de pedidos

Un cliente tendrá una lista de pedidos, la cuál podrá examinar y ver cada pedido con su correspondiente estado.

[Stereotype is «Functional»]

SCENARIOS

Basic Path. Darse de baja

SCENARIOS

El cliente podrá darse de baja a si mismo sólo si no tiene pedidos pendientes.

1. EL usuario apreta en la opcion de darse de baja y confirma utilizando su contraseña.

2. El sistema revisa que la contraseña sea correcta y que éste no tenga pedidos pendientes. Si todo está bien, efectua un log out del cliente y borra su cuenta del sistema. Posterior a ello, muestra un mensaje de que la cuenta fue correctamente eliminada

Exception: 2a. El cliente tiene pedidos en proceso en su cuenta

Exception: 2b. El usuario introduce mal su contraseña

 Exception. El usuario introduce mal su contraseña

1. El sistema marca el cuadro de contraseña y muestra un mensaje de que la contraseña es incorrecta y lo intente nuevamente.

 Exception. El cliente tiene pedidos en proceso en su cuenta

1. El sistema muestra un mensaje diciendo que debe cancelar o esperar a que los pedidos sean finalizados antes de poder darse de baja

CONSTRAINTS

 Pre-condition. Cliente logueado

El usuario deberá estar logueado como un cliente.

[Approved, Weight is 0.]

 Process. No hay pedidos sin terminar

El cliente no ha de tener pedidos sin finalizar / cancelar antes de poder borrar su cuenta

[Approved, Weight is 1.]

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_005 Darse de baja to «Functional» RF_017 Baja de cliente pública

[Direction is 'Source -> Destination'.]

 Realization from CU_005 Darse de baja to «Functional» RF_020 Lista de pedidos

[Direction is 'Source -> Destination'.]

CONNECTORS

 UseCaseLink Source -> Destination

From: Cliente : Actor, Public

To: CU_005 Darse de baja : UseCase, Public

CU_006 Modificación de Personal

UseCase in package 'Use Case Model'

El usuario dispondrá de una interfaz como la del alta del personal pero con la información del empleado seleccionado y un botón para confirmar los cambios realizados.

CU_006 Modificación de Personal
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 14/10/2017. Last modified 24/10/2017
Alias Gestión de Personal
Keywords CU_006

EXTERNAL REQUIREMENTS

- Requirement. RF_033 Gestión de personal

El administrador podrá dar de alta a nuevos miembros del staff. Así como modificarlos y darlos de baja.

[Stereotype is «Functional»]

SCENARIOS

- Basic Path. Modificar Personal

1. El sistema muestra una vista con la información del empleado seleccionado y los campos habilitados para su edición.
2. El usuario modifica algún (o ningún) campo y apreta en modificar.
Exception: 2a. El usuario ingresa un dato erróneo
3. El sistema revisa que los datos estén ingresados correctamente y luego realiza los cambios en la base de datos. Mostrando un mensaje de que el personal se ha modificado correctamente.

- Exception. El usuario ingresa un dato erróneo

1. El sistema marcará los campos erróneos y un mensaje de que hay campos vacíos o con datos incorrectos.

CONSTRAINTS

- Pre-condition. El usuario está logueado como administrador.

El administrador deberá estar logueado para que pueda acceder.

[Approved, Weight is 0.]

OUTGOING STRUCTURAL RELATIONSHIPS

- Realization from CU_006 Modificación de Personal to «Functional» RF_033 Gestión de personal

[Direction is 'Source -> Destination'.]

CONNECTORS

- Extend «extend» Source -> Destination

From: CU_006 Modificación de Personal : UseCase, Public
To: CU_021 Ver listado del personal : UseCase, Public

CU_007 Alta de envases

UseCase in package 'Use Case Model'

El usuario será capaz de agregar nuevos tipos de envases, especificando su nombre, descripción, capacidad, y un coeficiente que actúa como multiplicador del precio final.

CU_007 Alta de envases
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 14/10/2017. Last modified 24/10/2017
Alias Alta de envases
Keywords CU_007

EXTERNAL REQUIREMENTS

- Requirement. RF_003 Alta y modificacion de envases

El administrador se encargará del alta de los envases , indicando su nombre, capacidad, imagen y el coeficiente del mismo.
Podrá también modificar un envase de la misma manera.

[Stereotype is «Functional»]

SCENARIOS

- Basic Path. Alta de envases

El usuario será capaz de agregar nuevos tipos de envases, especificando su nombre, descripción, capacidad, y un coeficiente que actúa como multiplicador del precio final.

1. El usuario llena los campos correspondientes al envase y presiona en el botón de agregar envase.
2. El sistema revisa que los datos estén correctamente escritos y luego busca si no existe en la base de datos. Si todo es correcto, mostrará un mensaje de que el envase fue agregado correctamente.

Exception: 2a. El envase ya existe en la base de datos

Exception: 2b. Hay campos vacíos o incorrectos

- Exception. El envase ya existe en la base de datos

1. El sistema muestra un mensaje diciendo que el envase con ese nombre ya existe en el sistema.

- Exception. Hay campos vacíos o incorrectos

1. El sistema marca aquellos campos que posean datos erróneos y un mensaje diciendo que hay campos vacíos o incorrectos.

CONSTRAINTS

- Pre-condition. Administrador logueado

El administrador deberá estar logueado para poder realizar las operaciones

[Approved, Weight is 0.]

OUTGOING STRUCTURAL RELATIONSHIPS

- Realization from CU_007 Alta de envases to «Functional» RF_003 Alta y modificacion de envases

[Direction is 'Source -> Destination'.]

CONNECTORS

- UseCaseLink Source -> Destination

From: Administrador : Actor, Public

To: CU_007 Alta de envases : UseCase, Public

CU_008 Alta de sucursales.

UseCase in package 'Use Case Model'

El usuario deberá encargarse del alta, baja y modificación de las sucursales de la empresa.

CU_008 Alta de sucursales.
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 15/10/2017. Last modified 24/10/2017
Alias Alta de sucursales
Keywords CU_008

EXTERNAL REQUIREMENTS

- Requirement. RF_007 Gestión de sucursales

El **administrador** deberá encargarse del alta de las sucursales, indicando su ubicación, localidad, nombre y un teléfono de contacto. Podrá modificar los datos de las demás sucursales y también dar de baja una sucursal de ser necesario.

[Stereotype is «Functional»]

- Requirement. RF_008 Geolocalización de las sucursales.

Se espera poder acceder a un mapa de las sucursales por medio de **Google Maps**

[Stereotype is «Functional»]

SCENARIOS

-  Basic Path. Alta de sucursal

El usuario podrá dar de alta una sucursal indicando su nombre y su ubicación geográfica por medio de **Google Maps**.

1. El usuario introduce el nombre de la sucursal en un cuadro de texto y toda la información correspondiente a la dirección de la misma.
2. El sistema le otorgará una posición en el mapa mediante geolocalización de los datos ingresados y buscará en la base de datos si la sucursal ya está cargada en el sistema y de no estarla la da de alta.

Exception: 2a. La sucursal ya existe en el sistema

-  Exception. La sucursal ya existe en el sistema

1. El sistema muestra un mensaje de error diciendo que aquella sucursal ya existe en el sistema.

CONSTRAINTS

-  Pre-condition. Administrador logueado.

El usuario deberá estar logueado con credenciales de administrador

[Approved, Weight is 0.]

OUTGOING STRUCTURAL RELATIONSHIPS

-  Realization from CU_008 Alta de sucursales. to «Functional» RF_007 Gestión de sucursales

[Direction is 'Source -> Destination'.]

OUTGOING STRUCTURAL RELATIONSHIPS

➡ Realization from CU_008 Alta de sucursales. to «Functional» RF_008 Geolocalización de las sucursales.
 [Direction is 'Source -> Destination'.]

CONNECTORS

↗ **UseCaseLink** Source -> Destination
 From: Administrador : Actor, Public
 To: CU_008 Alta de sucursales. : UseCase, Public

CU_009 Agregar cerveza al carrito.

UseCase in package 'Use Case Model'

El usuario puede comprar una cerveza definiendo el tipo de cerveza que quiera (seleccionando de una lista). Y posterior a ello dispondrá de una lista con los envases disponibles para seleccionar, así como un cuadro de texto en donde introducir la cantidad que quiere de ese producto en ese envase. También tendrá a disposición el precio unitario del producto y el subtotal en base a la cantidad que desea. Junto a un botón para agregar al carro.

CU_009 Agregar cerveza al carrito.
 Version 1.0 Phase 1.0 Proposed
 Rodrigo Soria created on 17/10/2017. Last modified 23/11/2017
 Alias Agregar cerveza al carrito.
 Keywords CU_009

EXTERNAL REQUIREMENTS

Requirement. RF_011 Calculo del valor de los productos

Se deberá calcular el valor de un producto (**envase+cerveza * cant**) De manera automática de acuerdo al precio por litro de la cerveza y un coeficiente del envase. Éste ultimo cumple con la función de aumentar o disminuir el precio final del producto.

[Stereotype is «Functional»]

Requirement. RF_012 agregar producto al carrito

Un cliente deberá ser capaz de seleccionar un tipo de cerveza, con un envase a elección y la cantidad que deseé para agregar a su carrito. Creando así varias líneas de pedidos.

Se considera linea de pedido a: **Cerveza + Envase + Cantidad + subtotal**.

[Stereotype is «Functional»]

Requirement. RF_013 Calculo del precio final del pedido

Los pedidos deberán tener el precio final de la suma de todos los productos que el cliente haya seleccionado (*ver RF_012*). Y si se desea (opcional) alguna especie de descuento.

[Stereotype is «Functional»]

SCENARIOS

 Basic Path. Comprar una cerveza

1. El usuario selecciona un tipo de cerveza de una lista con imágenes de las cervezas existentes.
2. El sistema carga una vista con la imagen, nombre y descripción de ese tipo de cerveza, a su vez también se mostrará una lista con los envases disponibles, el precio unitario del producto (envase + cerveza), un cuadro de texto para indicar la cantidad

SCENARIOS

- que se desea de ese producto, un cuadro que muestra el subtotal (producto * cantidad). Y, si el usuario está registrado, un botón para agregar al carrito. Si no, uno para iniciar sesión.
3. El usuario indica la cantidad del producto que desee y Presiona el botón "agregar al carrito".
 4. El sistema muestra un mensaje de que el producto/s se han agregado correctamente al carrito.

CONSTRAINTS

 Pre-condition. Cliente logueado

El usuario debe estar logueado como cliente en el sistema para que éste puede acceder a las funciones.

[Approved, Weight is 0.]

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_009 Agregar cerveza al carrito. to «Functional» RF_011 Calculo del valor de los productos
[Direction is 'Source -> Destination'.]

 Realization from CU_009 Agregar cerveza al carrito. to «Functional» RF_012 agregar producto al carrito
[Direction is 'Source -> Destination'.]

 Realization from CU_009 Agregar cerveza al carrito. to «Functional» RF_013 Calculo del precio final del pedido
[Direction is 'Source -> Destination'.]

CONNECTORS

 Extend «extend» Source -> Destination
From: CU_009 Agregar cerveza al carrito. : UseCase, Public
To: CU_026 Ver productos : UseCase, Public

 Extend «extend» Source -> Destination
From: CU_009 Agregar cerveza al carrito. : UseCase, Public
To: CU_025 Ir a inicio : UseCase, Public

 UseCaseLink Source -> Destination
From: Cliente : Actor, Public
To: CU_009 Agregar cerveza al carrito. : UseCase, Public

CU_010 Ver carrito y confirmar compra

UseCase in package 'Use Case Model'

El usuario podrá ver el estado del carrito de compras actual, con la información perteneciente a todos los productos que él haya añadido, así como los subtotales correspondientes y el saldo total de todos los productos.
Y podrá presionar el botón de confirmar compra para que el pedido sea cargado en el sistema.

CU_010 Ver carrito y confirmar compra
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 18/10/2017. Last modified 22/11/2017
Alias Confirmar compra

EXTERNAL REQUIREMENTS

Requirement. RF_010 Carrito de compras "Pedido"

El Cliente tendrá un carrito de compras llamado "Pedido". El cual tendrá todos los productos que el cliente haya solicitado. almacenando cada producto como una "línea de pedido" (*ver RF_012*).

[Stereotype is «Functional»]

Requirement. RF_014 Envio

Un pedido puede tener un envío específico a una determinada ubicación. Por lo que la implementación de los mismos es necesaria. Si el usuario no especifica la información, se tomará como destino por defecto la dirección del cliente mismo (A menos que sea retiro en sucursal) (*ver RF_019*).

[Stereotype is «Functional»]

Requirement. RF_015 Compra

El cliente, una vez esté satisfecho con su pedido (*ver RF_016*) podrá realizar la operación de compra. Y se creará un registro con el pedido ya realizado, o como un pedido en espera de envío. (*ver RF_019 y RF_018*).

[Stereotype is «Functional»]

Requirement. RF_016 Verificación de pedido

El cliente deberá poder revisar su pedido antes de apretar el botón para que se proceda la compra, dándole la oportunidad de cancelar líneas de pedido que haya creado.

El cliente deberá seleccionar su método de envío preferido (*ver RF_019*).

[Stereotype is «Functional»]

Requirement. RF_018 Clasificacion de pedidos

el cliente tendrá pedidos, y los pedidos tienen un estado, el cual puede ser o solicitado, o en proceso, o enviado o finalizado.

[Stereotype is «Functional»]

Requirement. RF_019 Metodo de envío

El cliente deberá poder optar entre realizar un envío a una determinada **ubicación específica**, elegir si **recibir el envío en su domicilio**, o realizar una **recarga por sucursal**.

Para ésta última tendrá que seleccionar la sucursal deseada desde **google maps** o un menú desplegable

[Stereotype is «Functional»]

SCENARIOS

Basic Path. Basic Path

1. El usuario presiona en el botón de revisar carrito de compras

2. El sistema carga una vista con una lista de todos los productos adquiridos (líneas de pedido). Los subtotales y el saldo final correspondiente. Así como botones para eliminar cada línea.

Exception: 2a. No hay productos en el carrito

Alternate: 2b. El usuario elimina una línea de pedido

3. El usuario presiona en el botón confirmar compra

4. El sistema carga una ventana la cual contendrá 2 botones para definir el tipo de envío. Uno de ellos es de "retiro en sucursal", y el otro es de entrega a domicilio, con la dirección del cliente tomada por defecto, pero con la posibilidad de modificarla.

Alternate: 4a. El usuario selecciona "retiro por sucursal"

SCENARIOS
<p>Alternate: 4b. El usuario selecciona una dirección específica</p> <p>5. El usuario selecciona entrega a domicilio con la dirección por defecto, o escribe una que desee, y presiona en el botón de confirmar</p> <p>6. El sistema archiva el pedido y le muestra al usuario un mensaje de que la compra fue realizada con éxito.</p>
<p> Exception. No hay productos en el carrito</p> <p>1. El sistema muestra un mensaje indicando que no hay productos en el carrito.</p>
<p> Alternate. El usuario elimina una linea de pedido</p> <p>1. La linea de pedido seleccionada para eliminar simplemente desaparece, y se re-ordena la lista y ajustan los precios.</p>
<p> Alternate. El usuario selecciona "retiro por sucursal"</p> <p>1. Se activa un menú desplegable en el cual podremos elegir la sucursal en la que queremos retirar el pedido.</p> <p>2. El usuario selecciona la fábrica que deseé y apreta en confirmar comprar.</p> <p>3. El sistema archiva el pedido y le muestra al usuario un mensaje de que la compra fue realizada con éxito.</p>
<p> Alternate. El usuario selecciona una dirección específica</p> <p>1. El sistema muestra un campo de texto para indicar la dirección del lugar en el que espera el pedido</p> <p>2. El usuario ingresa la dirección y presiona en el botón "confirmar compra"</p> <p>3. El sistema archiva el pedido y le muestra al usuario un mensaje de que la compra fue realizada con éxito.</p>
OUTGOING STRUCTURAL RELATIONSHIPS
<p> Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_018 Clasificacion de pedidos [Direction is 'Source -> Destination'.]</p>
<p> Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_016 Verificación de pedido [Direction is 'Source -> Destination'.]</p>
<p> Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_014 Envio [Direction is 'Source -> Destination'.]</p>
<p> Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_015 Compra [Direction is 'Source -> Destination'.]</p>
<p> Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_010 Carrito de compras "Pedido" [Direction is 'Source -> Destination'.]</p>
<p> Realization from CU_010 Ver carrito y confirmar compra to «Functional» RF_019 Metodo de envío [Direction is 'Source -> Destination'.]</p>
CONNECTORS

CONNECTORS

 **UseCaseLink** Source -> Destination
 From: Cliente : Actor, Public
 To: CU_010 Ver carrito y confirmar compra : UseCase, Public

CU_011 Ver perfil

UseCase in package 'Use Case Model'

El usuario puede ver su información básica y un botón para modificar sus datos.

CU_011 Ver perfil
 Version 1.0 Phase 1.0 Proposed
 Rodrigo created on 18/10/2017. Last modified 20/11/2017
 Alias Ver perfil
 Keywords CU_011

EXTERNAL REQUIREMENTS

Requirement. RF_032 Ver perfil Cliente

El usuario deberá ser capaz de ver su perfil con la información de su cuenta.
 también debería ser capaz de acceder a una lista con los pedidos que haya realizado. (historial). (*ver RF_020*).

[Stereotype is «Functional»]

SCENARIOS

 Basic Path. Basic Path

1. El usuario apreta en el botón de perfil y accede a su información básica.
2. El sistema carga los datos correspondientes al cliente, su información básica y un botón de modificar cliente
 Alternate: 2a. El usuario apreta en el botón de modificar

 Alternate. El usuario apreta en el botón de modificar

1. El sistema carga una ventana como la del alta de cliente pero con la información del cliente seleccionado, los campos habilitados para su edición, y un botón para confirmar los cambios. (VER C.U. 004)

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_011 Ver perfil to «Functional» RF_032 Ver perfil Cliente

[Direction is 'Source -> Destination'.]

CONNECTORS

 **UseCaseLink** Source -> Destination
 From: Cliente : Actor, Public
 To: CU_011 Ver perfil : UseCase, Public

CONNECTORS
 Extend «extend» Source -> Destination From: CU_004 Modificación de Clientes : UseCase, Public To: CU_011 Ver perfil : UseCase, Public

CU_012 Ver listado de cervezas

UseCase in package 'Use Case Model'

El usuario podrá ver el listado de todas las cervezas cargadas en el sistema, así como botones de eliminar y modificar. Y un botón para definir el estado de la misma (si está activa o inactiva);

CU_012 Ver listado de cervezas
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 18/10/2017. Last modified 20/10/2017
Alias Ver listado de cervezas
Keywords CU_012

EXTERNAL REQUIREMENTS
<input checked="" type="checkbox"/> Requirement. RF_002 Eliminado de cervezas <p>El administrador podrá borrar un tipo de cerveza.</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>
<input checked="" type="checkbox"/> Requirement. RF_022 Listado de cervezas <p>El administrador deberá ser capaz de ver un listado con los tipos de cerveza de la base de datos así como diversos botones para modificar, eliminar y cambiar el estado de las mismas. (ver RF_027)</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>
<input checked="" type="checkbox"/> Requirement. RF_027 Estado de cervezas y envases. <p>tanto la cerveza como los envases, al ser productos para la venta, deberán tener un estado; es decir, si están habilitados o deshabilitados para la venta.</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>

SCENARIOS
 Basic Path. Basic Path <p>1. El usuario presiona en un botón para ver la lista de cervezas cargadas en el sistema. 2. El sistema carga una vista con la imagen de la cerveza y la información básica de la misma. Así como dos botones principales, uno de eliminar y otro de modificar. Y un botón que la define como activa o inactiva.</p> <p>Alternate: 2a. El usuario presiona en el botón de cambiar estado Alternate: 2b. El usuario presiona en el botón de eliminar cerveza Alternate: 2c. El usuario presiona en el botón de modificar cerveza</p>
 Alternate. El usuario presiona en el botón de cambiar estado <p>1. El sistema recarga la página y el estado de la cerveza cambia de activo a inactivo, o viceversa.</p>

SCENARIOS

 Alternate. El usuario presiona en el botón de eliminar cerveza

1. El sistema muestra una ventana con la información de la cerveza y su correspondiente imagen. Junto con un mensaje de si se está seguro de que quiere borrar a la misma. Y un botón para confirmar la eliminación, así como otro para cancelarla.
2. El usuario presiona en el botón de confirmar borrado de la misma.
3. El sistema elimina de la base de datos la cerveza seleccionada y recarga la página del listado.

 Alternate. El usuario presiona en el botón de modificar cerveza

1. El sistema carga una página como la del alta de cervezas pero con los campos habilitados para la edición, así como un botón para confirmar los cambios. Ver CU_013

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_012 Ver listado de cervezas to «Functional» RF_027 Estado de cervezas y envases.

[Direction is 'Source -> Destination'.]

 Realization from CU_012 Ver listado de cervezas to «Functional» RF_022 Listado de cervezas

[Direction is 'Source -> Destination'.]

 Realization from CU_012 Ver listado de cervezas to «Functional» RF_002 Eliminado de cervezas

[Direction is 'Source -> Destination'.]

CONNECTORS

 Extend «extend» Source -> Destination

From: CU_013 Modificar cerveza : UseCase, Public
To: CU_012 Ver listado de cervezas : UseCase, Public

 UseCaseLink Source -> Destination

From: Administrador : Actor, Public
To: CU_012 Ver listado de cervezas : UseCase, Public

CU_013 Modificar cerveza

UseCase in package 'Use Case Model'

El usuario puede modificar una cerveza del sistema

CU_013 Modificar cerveza

Version 1.0 Phase 1.0 Proposed

Rodrigo created on 18/10/2017. Last modified 20/10/2017

Alias Modificar cerveza

Keywords CU_013

EXTERNAL REQUIREMENTS

EXTERNAL REQUIREMENTS
<p> Requirement. RF_001 Alta y modificación de cervezas</p> <p>El administrador se encargará del alta de los tipos de cerveza, indicando su nombre, descripción, una imagen y el precio por litro de la misma. Podrá también modificar un tipo de cerveza de la misma manera.</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. El sistema carga una vista como la del alta de cervezas, con la información de la cerveza seleccionada y los campos habilitados para su edición. Así como también un botón para confirmar los cambios 2. El usuario realiza modificaciones (o no) en los campos y presiona en el botón de confirmar cambios 3. El sistema verifica los datos, registra los cambios en la base de datos y recarga la página del listado de cerveza <p>Exception: 3a. Ingreso mal de datos o campos vacíos.</p>
<p> Exception. Ingreso mal de datos o campos vacíos.</p> <ol style="list-style-type: none"> 1. El sistema marca los campos erróneos y muestra un mensaje de que están mal ingresados o vacíos.

OUTGOING STRUCTURAL RELATIONSHIPS
<p> Realization from CU_013 Modificar cerveza to «Functional» RF_001 Alta y modificación de cervezas</p> <p style="text-align: right;">[Direction is 'Source -> Destination'.]</p>

CONNECTORS
<p> Extend «extend» Source -> Destination</p> <p>From: CU_013 Modificar cerveza : UseCase, Public</p> <p>To: CU_012 Ver listado de cervezas : UseCase, Public</p>

CU_014 Modificar sucursal

UseCase in package 'Use Case Model'

El usuario tendrá a su disposición una vista como la del alta de sucursal, pero con la información de la sucursal seleccionada y un botón para confirmar los cambios.

CU_014 Modificar sucursal
 Version 1.0 Phase 1.0 Proposed
 usuario created on 20/10/2017. Last modified 20/10/2017
 Alias Modificar sucursal
 Keywords CU_014

EXTERNAL REQUIREMENTS

EXTERNAL REQUIREMENTS
<p> Requirement. RF_007 Gestión de sucursales</p> <p>El administrador deberá encargarse del alta de las sucursales, indicando su ubicación, localidad, nombre y un teléfono de contacto. Podrá modificar los datos de las demás sucursales y también dar de baja una sucursal de ser necesario.</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>
<p> Requirement. RF_008 Geolocalización de las sucursales.</p> <p>Se espera poder acceder a un mapa de las sucursales por medio de Google Maps</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>
SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. El sistema carga una vista como la del alta de sucursales pero con la información de la misma y los campos habilitados para su edición. Así como un botón para confirmar los cambios 2. El usuario modifica algún o ningún campo y presiona en el botón de confirmar cambios. 3. El sistema revisa que los campos estén correctamente llenados y realiza los cambios en la base de datos. Exception: 3a. Campo incorrecto o vacío.
<p> Exception. Campo incorrecto o vacío.</p> <ol style="list-style-type: none"> 1. El sistema marca aquellos campos erróneos y muestra un mensaje de error diciendo que los campos están mal ingresados o vacíos.
OUTGOING STRUCTURAL RELATIONSHIPS
<p> Realization from CU_014 Modificar sucursal to «Functional» RF_007 Gestión de sucursales</p> <p style="text-align: right;">[Direction is 'Source -> Destination'.]</p>
<p> Realization from CU_014 Modificar sucursal to «Functional» RF_008 Geolocalización de las sucursales.</p> <p style="text-align: right;">[Direction is 'Source -> Destination'.]</p>
CONNECTORS
<p> Extend «extend» Source -> Destination</p> <p>From: CU_014 Modificar sucursal : UseCase, Public</p> <p>To: CU_019 Listado de sucursales : UseCase, Public</p>

CU_015 Eliminar un rol

UseCase in package 'Use Case Model'

El usuario estará frente a una vista con la información del rol seleccionado y un botón de si se está seguro de eliminar el rol.

CU_015 Eliminar un rol
Version 1.0 Phase 1.0 Proposed

Rodrigo created on 19/10/2017. Last modified 19/10/2017

Alias Eliminar un rol

Keywords CU_015

EXTERNAL REQUIREMENTS

- Requirement. RF_006 Gestión de roles

El **administrador** puede dar de alta un rol con su respectiva descripción, modificarlo o eliminar uno.

Para borrar un rol (*Ver RF_021*).

[Stereotype is «Functional»]

- Requirement. RF_021 Eliminación de roles

para poder eliminar un rol primero no deberá haber ninguna cuenta con el mismo. Se tendrá que asignar un rol diferente a aquellos que posean rol antes de proceder con la eliminación

[Stereotype is «Functional»]

SCENARIOS

-  Basic Path. Basic Path

1. El sistema carga una vista con la información del rol seleccionado y un botón de si está seguro de querer eliminar el rol.
2. El usuario presiona en el botón para confirmar el borrado
3. El sistema borra el rol de la base de datos, recarga la vista del listado de roles y muestra un mensaje de que el rol se ha borrado correctamente.

Alternate: 3a. Existen usuarios con ese rol

-  Alternate. Existen usuarios con ese rol

1. El sistema muestra un pop-up con un mensaje diciendo que existen usuarios que poseen el rol seleccionado. Y mostrará un cuadro de texto desplegable para que el usuario seleccione un rol de reemplazo para los mismos. Así como un botón para continuar con el borrado y otro para cancelar la operación
2. El usuario selecciona uno de los roles y clickea en el botón de continuar.
3. El sistema asigna el rol a quienes corresponda y luego elimina el rol previamente seleccionado del sistema. Recarga la página del listado de roles y muestra un mensaje de que los cambios han sido efectuados correctamente.

OUTGOING STRUCTURAL RELATIONSHIPS

-  Realization from CU_015 Eliminar un rol to «Functional» RF_006 Gestión de roles

[Direction is 'Source -> Destination'.]

-  Realization from CU_015 Eliminar un rol to «Functional» RF_021 Eliminación de roles

[Direction is 'Source -> Destination'.]

CONNECTORS

-  Extend «extend» Source -> Destination

From: CU_015 Eliminar un rol : UseCase, Public
To: CU_017 Ver listado de roles : UseCase, Public

CU_016 Modificar un rol

UseCase in package 'Use Case Model'

El usuario dispondrá de una ventana como la del alta de roles, pero con la información del rol seleccionado y los campos listos para la edición

CU_016 Modificar un rol
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 19/10/2017
Alias Modificar un rol
Keywords CU_016

EXTERNAL REQUIREMENTS

- Requirement. RF_006 Gestión de roles

El **administrador** puede dar de alta un rol con su respectiva descripción, modificarlo o eliminar uno.
Para borrar un rol (*Ver RF_021*).

[Stereotype is «Functional»]

SCENARIOS

-  Basic Path. Basic Path

1. El sistema carga una vista como la del alta de roles, pero con la información del rol seleccionado y los campos disponibles para su modificación.
2. El usuario modifica algún o ningún campo y presiona en el botón de confirmar cambios
3. El sistema verifica que todos los cambios se hayan realizado correctamente y modifica los datos del sistema. Posterior a eso recarga la página del listado de roles y muestra un mensaje de que los cambios se han realizado correctamente
Exception: 3a. Existen campos con datos incorrectos o incompletos

-  Exception. Existen campos con datos incorrectos o incompletos

1. El sistema marca los campos erróneos y muestra un mensaje de que están mal ingresados o vacíos.

OUTGOING STRUCTURAL RELATIONSHIPS

-  Realization from CU_016 Modificar un rol to «Functional» RF_006 Gestión de roles

[Direction is 'Source -> Destination'.]

CONNECTORS

-  Extend «extend» Source -> Destination
From: CU_016 Modificar un rol : UseCase, Public
To: CU_017 Ver listado de roles : UseCase, Public

CU_017 Ver listado de roles

UseCase in package 'Use Case Model'

El usuario deberá ser capaz de ver un listado de los roles existentes y botones para eliminar o modificar los mismos.

CU_017 Ver listado de roles

Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 19/10/2017
Alias Ver listado de roles
Keywords CU_017

EXTERNAL REQUIREMENTS

- Requirement. RF_024 Listado de roles

El administrador podrá ver una lista de los roles disponibles en la base de datos y botones para modificar o eliminar a los mismos.

[Stereotype is «Functional»]

SCENARIOS

-  Basic Path. Basic Path

1. El usuario presiona en el botón de listado de roles
2. El sistema carga una vista con una lista de todos los roles existentes, su nombre y descripción, y dos botones; uno de modificación, y otro de eliminación.

Alternate: 2a. El usuario presiona el botón de eliminar rol

Alternate: 2b. El usuario presiona el botón de modificar rol

-  Alternate. El usuario presiona el botón de modificar rol

1. El sistema carga una vista como la del alta de roles, pero con la información del rol seleccionado y los campos disponibles para su modificación. VER CU_016

-  Alternate. El usuario presiona el botón de eliminar rol

1. El sistema carga una vista con la información del rol seleccionado y un botón de si está seguro de querer eliminar el rol.
VER CU_015

OUTGOING STRUCTURAL RELATIONSHIPS

-  Realization from CU_017 Ver listado de roles to «Functional» RF_024 Listado de roles

[Direction is 'Source -> Destination'.]

CONNECTORS

-  Extend «extend» Source -> Destination

From: CU_015 Eliminar un rol : UseCase, Public
To: CU_017 Ver listado de roles : UseCase, Public

-  Extend «extend» Source -> Destination

From: CU_016 Modificar un rol : UseCase, Public
To: CU_017 Ver listado de roles : UseCase, Public

-  UseCaseLink Source -> Destination

From: Administrador : Actor, Public
To: CU_017 Ver listado de roles : UseCase, Public

CU_018 Ver listado de envases

UseCase in package 'Use Case Model'

El usuario accederá a un listado con todos los envases disponibles, dos botones principales, de eliminado y modificación, y un botón para cambiar el estado de la cerveza. (Si está activa o inactiva).

CU_018 Ver listado de envases
 Version 1.0 Phase 1.0 Proposed
 Rodrigo created on 19/10/2017. Last modified 20/10/2017
 Alias Ver listado de envases
 Keywords CU_018

EXTERNAL REQUIREMENTS
<p><input checked="" type="checkbox"/> Requirement. RF_004 Eliminado de Envases</p> <p>El administrador podrá eliminar un envase.</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>
<p><input checked="" type="checkbox"/> Requirement. RF_023 Listado de envases</p> <p>El administrador deberá ser capaz de ver un listado con los envases de la base de datos así como diversos botones para modificar, eliminar y cambiar el estado de los mismos. (ver RF_027)</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>
<p><input checked="" type="checkbox"/> Requirement. RF_027 Estado de cervezas y envases.</p> <p>tanto la cerveza como los envases, al ser productos para la venta, deberán tener un estado; es decir, si están habilitados o deshabilitados para la venta.</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. El usuario presiona en un botón para ver la lista de envases cargados en el sistema. 2. El sistema carga una vista con la imagen del envase y la información básica del mismo. Así como dos botones principales, uno de eliminar y otro de modificar. Y un botón que lo define como activo o inactivo. <ul style="list-style-type: none"> Alternate: 2a. El usuario presiona en el botón de cambiar estado del envase Alternate: 2b. El usuario presiona en el botón de modificación de envase Alternate: 2c. El usuario presiona en el botón de eliminar envase
<p> Alternate. El usuario presiona en el botón de cambiar estado del envase</p> <ol style="list-style-type: none"> 1. El sistema recarga la página y el estado del envase cambia de activo a inactivo, o viceversa.
<p> Alternate. El usuario presiona en el botón de modificación de envase</p> <ol style="list-style-type: none"> 1. El sistema muestra una ventana como la del alta de envases pero con la información del envase seleccionado, los campos habilitados para la edición y un botón de confirmar cambios VER CU_020
<p> Alternate. El usuario presiona en el botón de eliminar envase</p>

SCENARIOS

1. El sistema muestra una vista con la información del envase a borrar y un botón para confirmar o no la eliminación del mismo.
2. El usuario presiona en el botón de confirmar la eliminación del envase.
3. El sistema elimina el envase en la base de datos y recarga la página del listado

OUTGOING STRUCTURAL RELATIONSHIPS

- ↳ Realization from CU_018 Ver listado de envases to «Functional» RF_004 Eliminado de Envases
 [Direction is 'Source -> Destination'.]
- ↳ Realization from CU_018 Ver listado de envases to «Functional» RF_023 Listado de envases
 [Direction is 'Source -> Destination'.]
- ↳ Realization from CU_018 Ver listado de envases to «Functional» RF_027 Estado de cervezas y envases.
 [Direction is 'Source -> Destination'.]

CONNECTORS

- ↗ **UseCaseLink** Source -> Destination
 From: Administrador : Actor, Public
 To: CU_018 Ver listado de envases : UseCase, Public
- ↗ **Extend** «extend» Source -> Destination
 From: CU_020 Modificar envase : UseCase, Public
 To: CU_018 Ver listado de envases : UseCase, Public

CU_019 Listado de sucursales

UseCase in package 'Use Case Model'

El usuario dispondrá de una vista con el listado de las sucursales.
 Junto con un botón para modificar y otro para eliminar cada sucursal.

CU_019 Listado de sucursales
 Version 1.0 Phase 1.0 Proposed
 usuario created on 20/10/2017. Last modified 20/10/2017
 Alias Listado de sucursales
 Keywords CU_019

EXTERNAL REQUIREMENTS

- Requirement. RF_007 Gestion de sucursales

El **administrador** deberá encargarse del alta de las sucursales, indicando su ubicación, localidad, nombre y un teléfono de contacto. Podrá modificar los datos de las demás sucursales y también dar de baja una sucursal de ser necesario.

[Stereotype is «Functional»]

EXTERNAL REQUIREMENTS
<p> Requirement. RF_025 Listado de sucursales</p> <p>El administrador deberá poder visualizar un listado con las sucursales almacenadas en el sistema, así como botones para modificarlas o eliminarlas.</p> <p style="text-align: right;">[Stereotype is «Functional»]</p>
SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. El usuario presiona en el botón de listado de sucursales 2. El sistema carga una vista con las sucursales cargadas en el sistema. Así como también un botón para modificar y otro para eliminar las mismas. <ul style="list-style-type: none"> Alternate: 2a. El usuario presiona el botón de modificar sucursal Alternate: 2b. El usuario presiona el botón de eliminar sucursal
<p> Alternate. El usuario presiona el botón de modificar sucursal</p> <ol style="list-style-type: none"> 1. El sistema carga una vista como la del alta de sucursales pero con la información de la misma y los campos habilitados para su edición. Así como un botón para confirmar los cambios. Ver CU_014
<p> Alternate. El usuario presiona el botón de eliminar sucursal</p> <ol style="list-style-type: none"> 1. El sistema muestra una vista con la información de la sucursal a borrar y un botón para confirmar o no la eliminación de la misma. 2. El usuario presiona el botón de confirmar el borrado de la misma. 3. El sistema elimina la sucursal de la base de datos y recarga la página del listado.
OUTGOING STRUCTURAL RELATIONSHIPS
<p> Realization from CU_019 Listado de sucursales to «Functional» RF_007 Gestión de sucursales</p> <p style="text-align: right;">[Direction is 'Source -> Destination'.]</p>
<p> Realization from CU_019 Listado de sucursales to «Functional» RF_025 Listado de sucursales</p> <p style="text-align: right;">[Direction is 'Source -> Destination'.]</p>
CONNECTORS
<p> UseCaseLink Source -> Destination</p> <p>From: Administrador : Actor, Public</p> <p>To: CU_019 Listado de sucursales : UseCase, Public</p>
<p> Extend «extend» Source -> Destination</p> <p>From: CU_014 Modificar sucursal : UseCase, Public</p> <p>To: CU_019 Listado de sucursales : UseCase, Public</p>

CU_020 Modificar envase

UseCase in package 'Use Case Model'

El usuario dispondrá de una vista como la del alta de envases pero con los campos habilitados para su edición. Y un botón de confirmar cambios.

CU_020 Modificar envase
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 19/10/2017
Alias Modificar envase
Keywords CU_020

EXTERNAL REQUIREMENTS

- Requirement. RF_003 Alta y modificacion de envases

El administrador se encargará del alta de los envases , indicando su nombre, capacidad, imagen y el coeficiente del mismo. Podrá también modificar un envase de la misma manera.

[Stereotype is «Functional»]

SCENARIOS

- Basic Path. Basic Path

1. El sistema muestra una ventana como la del alta de envases pero con la información del envase seleccionado, los campos habilitados para la edición y un botón de confirmar cambios.
 2. El usuario modifica algún o ningún dato y presiona en el botón de confirmar cambios.
 3. el sistema revisa que todos los campos estén correctamente ingresados y de ser así realiza las modificaciones necesarias en la base de datos
- Exception: 3a. Hay campos con datos incorrectos o incompletos

- Exception. Hay campos con datos incorrectos o incompletos

1. El sistema marca los campos erróneos y muestra un mensaje de que están mal ingresados o vacíos.

OUTGOING STRUCTURAL RELATIONSHIPS

- Realization from CU_020 Modificar envase to «Functional» RF_003 Alta y modificacion de envases

[Direction is 'Source -> Destination'.]

CONNECTORS

- Extend «extend» Source -> Destination

From: CU_020 Modificar envase : UseCase, Public
To: CU_018 Ver listado de envases : UseCase, Public

CU_021 Ver listado del personal*UseCase in package 'Use Case Model'*

El usuario dispondrá de una lista con todos los empleados registrados Y botones para eliminar o modificar sus perfiles.

CU_021 Ver listado del personal
Version 1.0 Phase 1.0 Proposed
Rodrigo created on 19/10/2017. Last modified 24/10/2017

EXTERNAL REQUIREMENTS

Requirement. RF_026 Listado de usuarios y personal

Un administrador deberá ser capaz de acceder a una lista con todas las cuentas del sistema, eligiendo si desea ver al personal o a los clientes. Se desea que pueda acceder a la información del mismo. Así como también poder borrar o modificar la información de estos.

[Stereotype is «Functional»]

Requirement. RF_033 Gestión de personal

El administrador podrá dar de alta a nuevos miembros del staff. Así como modificarlos y darlos de baja.

[Stereotype is «Functional»]

SCENARIOS

Basic Path. Basic Path

1. El usuario presiona el botón de listado de usuarios
2. El sistema muestra una vista con la lista de todos los empleados registrados. Así como también un botón para modificar el perfil del mismo y otro de eliminar.
Alternate: 2a. El usuario presiona el botón de eliminar
Alternate: 2b. El usuario presiona el botón de modificar

Alternate. El usuario presiona el botón de eliminar

1. El sistema muestra una ventana con los datos del empleado seleccionado y un botón para confirmar o cancelar la eliminación.
2. El usuario presiona el botón de confirmar el borrado.
3. El sistema borra el empleado del sistema y recarga la página del listado.

Alternate. El usuario presiona el botón de modificar

1. El sistema muestra una vista con la información del empleado seleccionado y los campos habilitados para su edición. Ver CU_006

OUTGOING STRUCTURAL RELATIONSHIPS

Realization from CU_021 Ver listado del personal to «Functional» RF_026 Listado de usuarios y personal [Direction is 'Source -> Destination'.]

Realization from CU_021 Ver listado del personal to «Functional» RF_033 Gestión de personal [Direction is 'Source -> Destination'.]

CONNECTORS

Extend «extend» Source -> Destination

From: CU_006 Modificación de Personal : UseCase, Public
To: CU_021 Ver listado del personal : UseCase, Public

CONNECTORS

 **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public
 To: CU_021 Ver listado del personal : UseCase, Public

CU_022 Alta de personal

UseCase in package 'Use Case Model'

El usuario dispondrá de una vista en la que deberá ingresar todos los datos de un empleado. Deberá ingresar los datos correspondientes a su email, nombre de usuario, nombre, apellido, contraseña, dirección, localidad, teléfono, DNI y el rol del mismo

CU_022 Alta de personal
 Version 1.0 Phase 1.0 Proposed
 Rodrigo created on 19/10/2017. Last modified 20/10/2017
 Alias Alta de personal
 Keywords CU_022

EXTERNAL REQUIREMENTS

 Requirement. RF_033 Gestión de personal

El administrador podrá dar de alta a nuevos miembros del staff. Así como modificarlos y darlos de baja.

[Stereotype is «Functional»]

SCENARIOS

 Basic Path. Basic Path

1. El usuario ingresa los datos correspondientes al email, nombre de usuario, nombre, apellido, contraseña, dirección, localidad, teléfono, DNI, y el rol correspondiente.
2. El sistema corroborará los datos, de estar todo bien, buscará en la base de datos si la cuenta, el nombre de usuario y el DNI existen. De no existir crea la cuenta correspondiente y muestra un mensaje de que el usuario ha sido registrado con éxito.
 Exception: 2a. Hay algún campo mal ingresado o vacío.

 Exception. Hay algún campo mal ingresado o vacío.

1. El sistema marca los campos erróneos y muestra un mensaje de error indicando que hay campos incorrectos o vacíos.

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_022 Alta de personal to «Functional» RF_033 Gestión de personal

[Direction is 'Source -> Destination'.]

CONNECTORS

 **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public
 To: CU_022 Alta de personal : UseCase, Public

CU_023 Ver listado de clientes

UseCase in package 'Use Case Model'

El usuario podrá acceder a una vista con la lista de todos los clientes cargados en el sistema. Así como también un botón para modificarlo o eliminarlo.

CU_023 Ver listado de clientes
 Version 1.0 Phase 1.0 Proposed
 usuario created on 20/10/2017. Last modified 20/10/2017
 Alias Ver listado de clientes
 Keywords CU_023

EXTERNAL REQUIREMENTS

- Requirement. RF_005 Gestión de clientes

El **administrador** se puede encargar del alta de clientes. Podrá también modificar un usuario de ser necesario. Además podrá eliminar un usuario ya registrado pero sólo si este no tiene pedidos que aún no hayan sido finalizados/cancelados. (*Ver RF_020*).

[Stereotype is «Functional»]

- Requirement. RF_020 Lista de pedidos

Un cliente tendrá una lista de pedidos, la cuál podrá examinar y ver cada pedido con su correspondiente estado.

[Stereotype is «Functional»]

- Requirement. RF_026 Listado de usuarios y personal

Un administrador deberá ser capaz de acceder a una lista con todas las cuentas del sistema, eligiendo si desea ver al personal o a los clientes. Se desea que pueda acceder a la información del mismo. Así como también poder borrar o modificar la información de estos.

[Stereotype is «Functional»]

SCENARIOS

-  Basic Path. Basic Path

1. El usuario presiona en el botón de listado de clientes
2. El sistema muestra una lista de todos los clientes cargados en el sistema y dos botones, para eliminarlo o modificarlo.
 - Alternate: 2a. El usuario presiona en el botón de eliminar cliente
 - Alternate: 2b. El usuario presiona en el botón de modificar cliente

-  Alternate. El usuario presiona en el botón de eliminar cliente

1. El sistema muestra una ventana con la información del cliente, un botón para confirmar el borrado y otro para cancelar.
2. El usuario presiona en el botón de confirmar.
3. El sistema borra el cliente de la base de datos y recarga la página del listado.

-  Alternate. El usuario presiona en el botón de modificar cliente

1. El sistema carga una ventana como la del alta de cliente pero con la información del cliente seleccionado, los campos habilitados para su edición, y un botón para confirmar los cambios.

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Realization from CU_023 Ver listado de clientes to «Functional» RF_026 Listado de usuarios y personal	[Direction is 'Source -> Destination'.]
↳ Realization from CU_023 Ver listado de clientes to «Functional» RF_005 Gestión de clientes	[Direction is 'Source -> Destination'.]
↳ Realization from CU_023 Ver listado de clientes to «Functional» RF_020 Lista de pedidos	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Extend «extend» Source -> Destination	
From: CU_004 Modificación de Clientes : UseCase, Public	
To: CU_023 Ver listado de clientes : UseCase, Public	
↗ UseCaseLink Source -> Destination	
From: Administrador : Actor, Public	
To: CU_023 Ver listado de clientes : UseCase, Public	

CU_024 Administrar envases por cervezas

UseCase in package 'Use Case Model'

El administrador dispondrá de una lista de las cervezas en el sistema y presionando sobre ellas se desplegará una lista con los envases disponibles en el sistema. Permiéndole tildar o destildar aquellos envases que estén disponibles para la venta de la cerveza.

CU_024 Administrar envases por cervezas
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 20/11/2017. Last modified 20/11/2017
Alias Administrar envases por cervezas
Keywords CU_024

EXTERNAL REQUIREMENTS	
☒ Requirement. RF_028 Envases por Cerveza	Cada cerveza se ha de vender en determinados envases específicos, y no en cualquiera. Debido a que lógicamente, no se puede vender cualquier cerveza en cualquier envase. [Stereotype is «Functional»]

SCENARIOS	
↗ Basic Path. Basic Path	<ol style="list-style-type: none"> 1. El usuario presiona en el botón de Envases por cerveza 2. El sistema carga y muestra una lista de las cervezas. 3. El usuario hace click en una de las cervezas de la lista 4. se despliega una lista cargada con los envases disponibles del sistema. Permiéndole tildar o destildar aquellos envases que quiera que estén, o no, disponibles para la venta de la cerveza seleccionada.

SCENARIOS

5. El usuario tilda y destilda los envases que deseé y presiona en el botón de modificar
6. El sistema realiza los cambios correspondientes y recarga la página

CONSTRAINTS

Pre-condition. Administrador logueado

El administrador de sistema deberá estar logueado para poder realizar cambios.

[Approved, Weight is 0.]

Pre-condition. Existencia de productos

Deben existir previamente cervezas y envases cargados en el sistema

[Approved, Weight is 1.]

OUTGOING STRUCTURAL RELATIONSHIPS

Realization from CU_024 Administrar envases por cervezas to «Functional» RF_028 Envases por Cerveza
[Direction is 'Source -> Destination'.]

CONNECTORS

UseCaseLink Source -> Destination

From: Administrador : Actor, Public

To: CU_024 Administrar envases por cervezas : UseCase, Public

CU_025 Ir a inicio

UseCase in package 'Use Case Model'

Cualquier usuario, registrado o no, mediante el botón de inicio será llevado a la página principal del sistema.

CU_025 Ir a inicio
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias Ir a inicio
Keywords CU_025

EXTERNAL REQUIREMENTS

Requirement. RF_030 Página de inicio.

Debe de existir una página de inicio general, igual para todos los tipos de usuarios, sean empleados o no, o estén conectados o no, que prevea acceso a la mayoría de las funciones que un usuario pueda hacer.

A su vez, debe existir otra segunda página de inicio pero del lado del backend para cuando un administrador decida entrar.

[Stereotype is «Functional»]

SCENARIOS

SCENARIOS

Basic Path. Basic Path

1. El usuario apreta el botón de inicio (o acaba de acceder a la página)
2. El sistema muestra una barra de menú superior (que siempre estará) con un botón de home, productos, contacto y, en el caso de ser un usuario no logueado, de registro, y a la derecha de log in. En caso de ser un usuario logueado será un botón con un menú desplegable. En el medio de la página aparecerán 3 cervezas escogidas al azar con un botón para ver el producto. Y en el parte inferior una barra con el nombre de la empresa y botones para contacto, home y productos.
- Alternate: 2a. El usuario es un administrador del lado del backend
3. El usuario presiona en el botón home
4. La página se recarga

Alternate. El usuario es un administrador del lado del backend

1. El sistema muestra una barra de menú superior (que siempre estará) con un botón de home, y un menú desplegable para: personal, otro para roles, sucursales, clientes, cervezas y envases. Cada uno con dos opciones: dar de alta, y listado. A excepción del de cervezas, que también dispondrá de la opción de administrar los envases por cerveza y de verificar la cantidad de litros vendidos.

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_025 Ir a inicio to «Functional» RF_030 Página de inicio.

[Direction is 'Source -> Destination'.]

CONNECTORS

Extend «extend» Source -> Destination

From: CU_025 Ir a inicio : UseCase, Public
To: CU_025 Ir a inicio : UseCase, Public

Extend «extend» Source -> Destination

From: CU_025 Ir a inicio : UseCase, Public
To: CU_025 Ir a inicio : UseCase, Public

UseCaseLink Source -> Destination

From: Usuario no registrado : Actor, Public
To: CU_025 Ir a inicio : UseCase, Public

Extend «extend» Source -> Destination

From: CU_027 Iniciar sesión : UseCase, Public
To: CU_025 Ir a inicio : UseCase, Public

Extend «extend» Source -> Destination

From: CU_026 Ver productos : UseCase, Public
To: CU_025 Ir a inicio : UseCase, Public

Extend «extend» Source -> Destination

From: CU_009 Agregar cerveza al carrito. : UseCase, Public
To: CU_025 Ir a inicio : UseCase, Public

CU_026 Ver productos

UseCase in package 'Use Case Model'

El sistema redigire al usuario a una vista con los tipos de cerveza que haya en la base de datos habilitadas para su venta.

CU_026 Ver productos
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias Ver productos
Keywords CU_026

EXTERNAL REQUIREMENTS

- Requirement. RF_029 Lista de cervezas para la venta

Cualquier usuario, registrado o no, deberá ser capaz de acceder a una lista con las cervezas disponibles para la venta
 [Stereotype is «Functional»]

SCENARIOS

-  Basic Path. Basic Path

1. El sistema muestra una vista con una lista de todos los tipos de cerveza que hay en la base de datos diviendo en páginas de a 15 cervezas por página. Si hay más de 15, después de la lista de cervezas, aparecerá un botón para cambiar de página.
2. El usuario elige de entre las cervezas disponibles y apreta en "ver producto". Ver CU_009

OUTGOING STRUCTURAL RELATIONSHIPS

-  Realization from CU_026 Ver productos to «Functional» RF_029 Lista de cervezas para la venta
 [Direction is 'Source -> Destination'.]

CONNECTORS

 **Extend** «extend» Source -> Destination
 From: CU_026 Ver productos : UseCase, Public
 To: CU_025 Ir a inicio : UseCase, Public

 **Extend** «extend» Source -> Destination
 From: CU_009 Agregar cerveza al carrito. : UseCase, Public
 To: CU_026 Ver productos : UseCase, Public

CU_027 Iniciar sesión

UseCase in package 'Use Case Model'

Cualquier usuario puede iniciar sesión en el sistema

CU_027 Iniciar sesión
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias Iniciar sesión
Keywords CU_027

SCENARIOS
<p> Basic Path. Basic Path</p> <ol style="list-style-type: none"> 1. El usuario presiona en el botón de log in 2. El sistema carga una vista con dos cuadros de texto, uno para ingresar el email con el que se haya registrado y otro para ingresar la contraseña. Al final un botón para ingresar al sistema 3. El usuario ingresa su e-mail y su contraseña y presiona en el botón de ingresar 4. El sistema corrobora los datos con los que estén en la base de datos y si todo sale bien se redirige al usuario a la página de inicio. <p>Exception: 4a. El email no existe o la contraseña no coincide</p>
<p> Exception. El email no existe o la contraseña no coincide</p> <ol style="list-style-type: none"> 1. El sistema mostrará un mensaje por encima del cuadro de email diciendo que las credenciales con las que intenta acceder son erróneas.

CONNECTORS
<p> Extend «extend» Source -> Destination From: CU_027 Iniciar sesión : UseCase, Public To: CU_025 Ir a inicio : UseCase, Public</p>
<p> UseCaseLink Source -> Destination From: Usuario no registrado : Actor, Public To: CU_027 Iniciar sesión : UseCase, Public</p>

CU_028 Ver historial de pedidos

UseCase in package 'Use Case Model'

El cliente puede acceder a su historial de pedidos y ver el estado de cada uno de los pedidos que haya realizado.

CU_028 Ver historial de pedidos
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017
Alias Ver historial de pedidos
Keywords CU_028

EXTERNAL REQUIREMENTS
<p> Requirement. RF_018 Clasificación de pedidos el cliente tendrá pedidos, y los pedidos tienen un estado, el cual puede ser o solicitado, o en proceso, o enviado o finalizado. [Stereotype is «Functional»]</p>
<p> Requirement. RF_020 Lista de pedidos Un cliente tendrá una lista de pedidos, la cual podrá examinar y ver cada pedido con su correspondiente estado. [Stereotype is «Functional»]</p>

SCENARIOS
<p> Basic Path. Basic Path</p> <p>El usuario abre una página en la cual puede observar cada uno de los pedidos que ha realizado durante su trayectoria como usuario registrado. Y ver el estado en el que se encuentran.</p> <ol style="list-style-type: none"> 1. El usuario apreta en el botón de ver historial de pedidos 2. El sistema carga una vista con una lista de todos los pedidos que él ha realizado, la fecha de creación, el tipo de envío, el objetivo del envío y el monto final a pagar. Además del estado actual del pedido.
CONSTRAINTS
<p> Pre-condition. El cliente debe estar logueado o ser un administrador quien acceda al historial de un cliente [Approved, Weight is 0.]</p> <p> Pre-condition. El cliente debe existir [Approved, Weight is 1.]</p> <p> Pre-condition. El cliente debe tener pedidos realizados Caso contrario no se mostrarán los pedidos sino un mensaje [Approved, Weight is 2.]</p>
OUTGOING STRUCTURAL RELATIONSHIPS
<p> Realization from CU_028 Ver historial de pedidos to «Functional» RF_020 Lista de pedidos [Direction is 'Source -> Destination'.]</p> <p> Realization from CU_028 Ver historial de pedidos to «Functional» RF_018 Clasificacion de pedidos [Direction is 'Source -> Destination'.]</p>
CONNECTORS
<p> UseCaseLink Source -> Destination From: Cliente : Actor, Public To: CU_028 Ver historial de pedidos : UseCase, Public</p> <p> Extend «extend» Source -> Destination From: CU_029 consultar pedido : UseCase, Public To: CU_028 Ver historial de pedidos : UseCase, Public</p>

CU_029 consultar pedido

UseCase in package 'Use Case Model'

Un cliente puede acceder a la información de un pedido que haya realizado.
Si el que accede es un administrador, éste puede cambiar el estado del pedido.

CU_029 consultar pedido

EXTERNAL REQUIREMENTS

Requirement. RF_018 Clasificacion de pedidos

el cliente tendrá pedidos, y los pedidos tienen un estado, el cual puede ser o solicitado, o en proceso, o enviado o finalizado.

[Stereotype is «Functional»]

Requirement. RF_031 consultar pedido

El administrador puede consultar un pedido de un cliente y modificar su estado.

A su vez, un cliente puede consultar un pedido para ver qué es lo que en él compró y su estado, pero no puede realizar modificaciones.

[Stereotype is «Functional»]

SCENARIOS

 Basic Path. Cliente consulta pedido

El cliente puede acceder a la información de un pedido, pero no modificarla.

1. El usuario presiona en un pedido de la lista de pedidos

2. El sistema carga una vista con toda la información del pedido. Cada uno de los productos, el total, el tipo de envío y el destino del mismo. A si mismo también muestra el estado actual del pedido.

Alternate: 2a. Administrador consulta pedido de un cliente

 Alternate. Administrador consulta pedido de un cliente

El administrador consulta la información del pedido y puede modificar su estado.

1. El sistema carga una vista con toda la información del pedido. Cada uno de los productos, el total, el tipo de envío y el destino del mismo. A si mismo también muestra un botón con el estado actual del pedido.

2. El administrador presiona en el botón del estado del pedido

3. El sistema despliega una lista con los estados disponibles

4. El administrador selecciona un nuevo estado para el pedido

5. El sistema registra los cambios en la base de datos y recarga la página

CONSTRAINTS

 Invariant. Un cliente o un empleado está conectado

[Approved, Weight is 0.]

 Invariant. El pedido existe

[Approved, Weight is 1.]

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_029 consultar pedido to «Functional» RF_018 Clasificacion de pedidos

[Direction is 'Source -> Destination'.]

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from CU_029 consultar pedido to «Functional» RF_031 consultar pedido

[Direction is 'Source -> Destination'.]

CONNECTORS

 Extend «extend» Source -> Destination

From: CU_029 consultar pedido : UseCase, Public

To: CU_028 Ver historial de pedidos : UseCase, Public

CU_030 ver cantidad de litros vendidos

UseCase in package 'Use Case Model'

El administrador deberá ser capaz de ver la cantidad de litros vendidos entre dos fechas

CU_030 ver cantidad de litros vendidos

Version 1.0 Phase 1.0 Proposed

Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017

Alias ver cantidad de litros vendidos

Keywords CU_030

EXTERNAL REQUIREMENTS

 Requirement. RF_034 ver cantidad de litros vendidos por cerveza entre fechas

El administrador deberá ser capaz de ver cuantos litros se vendieron de determinada cerveza entre dos fechas específicas

[Stereotype is «Functional»]

SCENARIOS

 Basic Path. Basic Path

1. El administrador presiona en el botón de consultar litros vendidos

2. El sistema muestra dos cuadros de texto para introducir fechas y un menú desplegable para seleccionar el tipo de cerveza que se desea buscar

3. El administrador ingresa las fechas correspondientes al inicio y final, y elige la cerveza deseada.

4. El sistema revisa que las fechas sean correctas, que la de inicio sea antes que la del final, y revisa también en base a los pedidos buscando los litros que se vendieron de la cerveza seleccionada. Si todo sale bien, muestra la imagen de la cerveza seleccionada, su nombre, descripción, y la cantidad de litros que se vendieron entre esas fechas.

Exception: 4a. La cerveza nunca se vendió entre esas fechas

Exception: 4b. Fechas incorrectas

 Exception. La cerveza nunca se vendió entre esas fechas

1. muestra la imagen de la cerveza seleccionada, su nombre, descripción, y en donde iría la cantidad de litros vendidos directamente irá un "0 litros vendidos".

 Exception. Fechas incorrectas

1. El sistema muestra un mensaje indicando que las fechas son incorrectas. Que corrobore el orden (inicio-final) y que ingrese los valores correctamente.

OUTGOING STRUCTURAL RELATIONSHIPS

➡ Realization from CU_030 ver cantidad de litros vendidos to «Functional» RF_034 ver cantidad de litros vendidos por cerveza entre fechas

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ **UseCaseLink** Source -> Destination

From: Administrador : Actor, Public

To: CU_030 ver cantidad de litros vendidos : UseCase, Public

Table of Contents

Domain Model	3
Domain Model diagram	3
Config	3
Config diagram	3
Autoload	4
Request	4
Router	5
Controllers	6
Controllers diagram	6
AdminController	8
CervezaController	8
ClienteController	10
CuentaController	11
EnvaseController	12
FrontController	13
PedidoController	15
RolController	16
SucursalController	17
UsuarioController	18
Daos	20
Daos diagram	20
DB	20
DB diagram	20
Connection	21
DBCervezaDAO	21
DBClienteDAO	23
DBEnvaseDAO	24
DBEnvxcerDAO	25
DBLineaPedidoDAO	26
DBPedidoDAO	27
DBRolDAO	28
DBSucursalDAO	28
DBUsuarioDAO	29
SingletonAbstractDAO	30
iDao	31
Models	34
Models diagram	34
Cerveza	34
Cliente	36
Envase	37
Envio	38
Envio	38
LineaPedido	39
Pedido	40
Persona	42
Rol	44
Sucursal	46
Usuario	47
Utils	49

Utils diagram	49
Thumb	49

Domain Model

Package in package 'Model'

Domain Model
Version Phase 1.0 Proposed
usuario created on 11/10/2017. Last modified 11/10/2017

Domain Model diagram

Class diagram in package 'Domain Model'

Domain Model
Version 1.0
usuario created on 11/10/2017. Last modified 11/10/2017

Figure 1: Domain Model

Config

Package in package 'Domain Model'

Config
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

Config diagram

Class diagram in package 'Config'

Config
Version 1.0
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

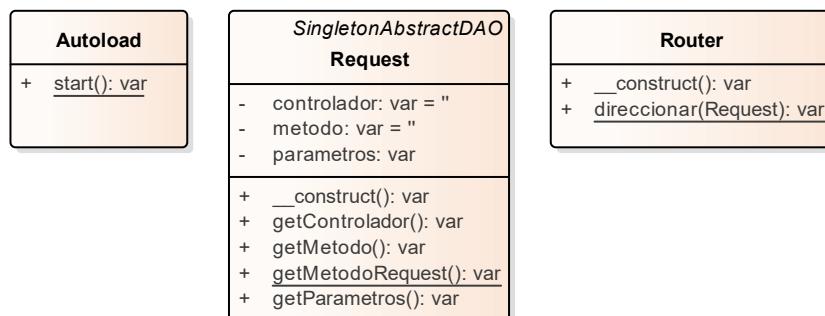


Figure 2: Config

Autoload

Class in package 'Config'

Autoload
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

OPERATIONS

◆ start () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Request

Class in package 'Config'

Request
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from Request to SingletonAbstractDAO

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ controlador : var Private = "

[Is static False. Containment is Not Specified.]

◆ metodo : var Private = "

[Is static False. Containment is Not Specified.]

◆ parametros : var Private

[Is static False. Containment is Not Specified.]

OPERATIONS

◆ __construct () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getControlador () : var Public

Devuelve el controlador
@return String

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ getMetodo () : var Public	
Devuelve el mÃ©todo @return String	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getMetodoRequest () : var Public	
Devuelve el mÃ©todo HTTP con el que se hizo el Request @return String	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getParametros () : var Public	
Devuelve los atributos @return Array	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Router

Class in package 'Config'

Router
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

OPERATIONS	
◆ __construct () : var Public	
Se encarga de direccionar a la pagina solicitada @param Request	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ direccionar (request : Request) : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Controllers

Package in package 'Domain Model'

Controllers
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

Controllers diagram

Class diagram in package 'Controllers'

Controllers
Version 1.0
Rodrigo Soria created on 21/11/2017. Last modified 22/11/2017

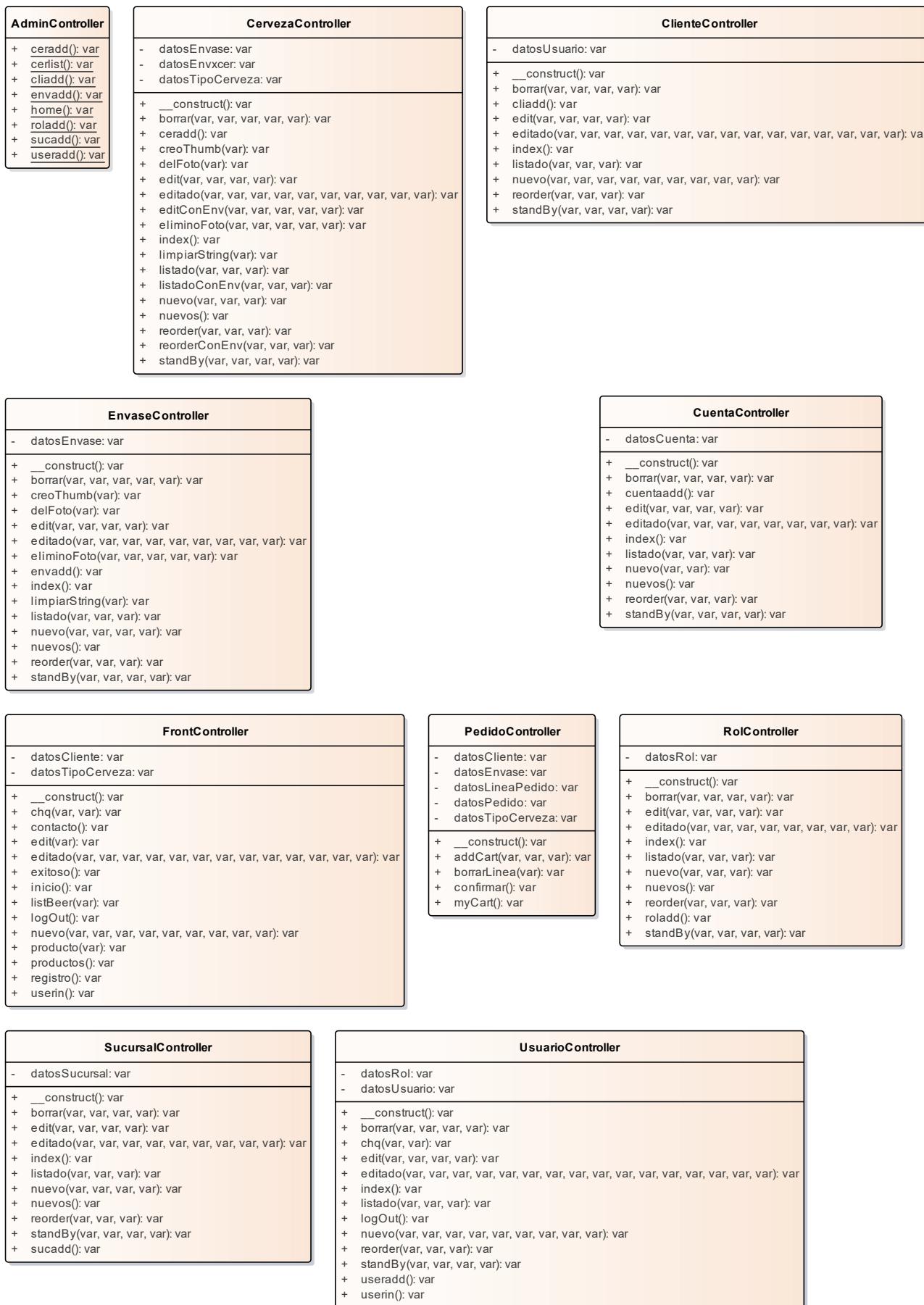


Figure 3: Controllers

AdminController

Class in package 'Controllers'

AdminController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

OPERATIONS	
◆ ceradd () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ cerlist () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ cliadd () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ envadd () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ home () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ roladd () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ sucadd () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ useradd () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

CervezaController

Class in package 'Controllers'

CervezaController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◆ datosEnvase : var Private	[Is static False. Containment is Not Specified.]
◆ datosEnvxcer : var Private	

ATTRIBUTES	
	[Is static False. Containment is Not Specified.]
◆ datosTipoCerveza : var Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ borrar (id : var , page : var , campo : var , orden : var , foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ ceradd () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ creoThumb (foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ delFoto (foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ edit (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ editado (cerveza : var , descripcion : var , precioXLitro : var , page : var , campo : var , orden : var , id : var , standBy : var , fotoOld : var , cervezaOld : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ editConEnv (envxcerCheck : var , page : var , campo : var , orden : var , idCerveza : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ eliminoFoto (id : var , page : var , campo : var , orden : var , foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ index () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ limpiaString (texto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listado (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listadoConEnv (page : var , campo : var , orden : var) : var Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ nuevo (cerveza : var , descripcion : var , precioXLitro : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nuevos () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reorder (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reorderConEnv (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ standBy (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

ClienteController

Class in package 'Controllers'

ClienteController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◆ datosUsuario : var Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ borrar (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ cliadd () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ edit (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ editado (nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var , idRol : var , page : var , campo : var , orden : var , id : var , standBy : var , contra : var) : var Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ index () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listado (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nuevo (nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var , idRol : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reorder (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ standBy (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

CuentaController

Class in package 'Controllers'

CuentaController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◆ datosCuenta : var Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ borrar (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ cuentaadd () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ edit (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ editado (cuenta : var , descripcion : var , permisos : var , page : var , campo : var , orden : var , id : var , standBy : var) :	

OPERATIONS	
var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ index () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ listado (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ nuevo (cuenta : var , descripcion : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ nuevos () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ reorder (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ standBy (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

EnvaseController

Class in package 'Controllers'

EnvaseController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◊ datosEnvase : var Private	[Is static False. Containment is Not Specified.]
OPERATIONS	
◊ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ borrar (id : var , page : var , campo : var , orden : var , foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ creoThumb (foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ delFoto (foto : var) : var Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ edit (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ editado (tipo : var , capacidad : var , coeficiente : var , page : var , campo : var , orden : var , id : var , standBy : var , fotoOld : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ eliminoFoto (id : var , page : var , campo : var , orden : var , foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ envadd () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ index () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ limpiarString (texto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listado (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nuevo (tipo : var , capacidad : var , coeficiente : var , foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ nuevos () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ reorder (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ standBy (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

FrontController

Class in package 'Controllers'

FrontController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	

ATTRIBUTES	
◆	datosCliente : var Private [Is static False. Containment is Not Specified.]
◆	datosTipoCerveza : var Private [Is static False. Containment is Not Specified.]
OPERATIONS	
◆	__construct () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	chq (user : var , pass : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	contacto () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	edit (id : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	editado (nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var , idRol : var , id : var , standBy : var , contra : var , emailOld : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	exitoso () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	inicio () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	listBeer (page : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	logOut () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	nuevo (nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var , idRol : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	producto (id : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	productos () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ registro () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ userin () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

PedidoController

Class in package 'Controllers'

PedidoController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◆ datosCliente : var Private	[Is static False. Containment is Not Specified.]
◆ datosEnvase : var Private	[Is static False. Containment is Not Specified.]
◆ datosLineaPedido : var Private	[Is static False. Containment is Not Specified.]
◆ datosPedido : var Private	[Is static False. Containment is Not Specified.]
◆ datosTipoCerveza : var Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addCart (cantidad : var , idCerveza : var , idEnvase : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ borrarLinea (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ confirmar () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

◆ myCart () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

RolController

Class in package 'Controllers'

RolController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES

◆ datosRol : var Private

[Is static False. Containment is Not Specified.]

OPERATIONS

◆ __construct () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ borrar (id : var , page : var , campo : var , orden : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ edit (id : var , page : var , campo : var , orden : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ editado (rol : var , descripcion : var , permisos : var , page : var , campo : var , orden : var , id : var , standBy : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ index () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ listado (page : var , campo : var , orden : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ nuevo (rol : var , descripcion : var , permisos : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ nuevos () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ reorder (page : var , campo : var , orden : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

◆ roladd () : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ standBy (id : var , page : var , campo : var , orden : var) : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

SucursalController*Class in package 'Controllers'*

SucursalController
 Version 1.0 Phase 1.0 Proposed
 Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES

◆ datosSucursal : var Private
 [Is static False. Containment is Not Specified.]

OPERATIONS

◆ __construct () : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ borrar (id : var , page : var , campo : var , orden : var) : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ edit (id : var , page : var , campo : var , orden : var) : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ editado (nombre : var , domicilio : var , localidad : var , telefono : var , page : var , campo : var , orden : var , id : var , standBy : var) : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ index () : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ listado (page : var , campo : var , orden : var) : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ nuevo (nombre : var , domicilio : var , localidad : var , telefono : var) : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ nuevos () : var Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ reorder (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ standBy (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ sucadd () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

UsuarioController

Class in package 'Controllers'

UsuarioController
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◆ datosRol : var Private	[Is static False. Containment is Not Specified.]
◆ datosUsuario : var Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ borrar (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ chq (user : var , pass : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ edit (id : var , page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ editado (nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var , idRol : var , page : var , campo : var , orden : var , id : var , standBy : var , contra : var , emailOld : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ index () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆	listado (page : var , campo : var , orden : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	logOut () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	nuevo (nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var , idRol : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	reorder (page : var , campo : var , orden : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	standBy (id : var , page : var , campo : var , orden : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	useradd () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	userin () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Daos

Package in package 'Domain Model'

Daos
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

Daos diagram

Class diagram in package 'Daos'

Daos
Version 1.0
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

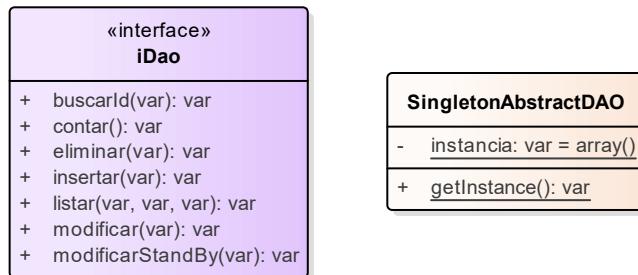


Figure 4: Daos

DB

Package in package 'Daos'

DB
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

DB diagram

Class diagram in package 'DB'

DB
Version 1.0
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

Connection	SingletonAbstractDAO DBCervezaDAO	SingletonAbstractDAO DBClienteDAO	SingletonAbstractDAO DBEnvaseDAO	SingletonAbstractDAO DBEnvxcerDAO
+ Connect(): var	# table: var = 'cervezas' + __construct(): var + buscarId(var): var + buscarTipo(var): var + contarActivos(): var + insertar(var): var + listAll(): var + listar(var, var, var): var + listarConEnv(var, var, var): var + listarConEnv1(var, var, var): var + listarConEnv2(var, var, var): var + listarSinStandBy(var): var + listarSinStandByLimit(): var + modificar(var): var + productoConEnv(var, var, var, var): var	# table: var = 'usuarios' + __construct(): var + buscarEmail(var): var + buscarId(var): var + buscarTipo(var): var + contarCliente(): var + insertar(var): var + listar(var, var, var): var + modificar(var): var	# table: var = 'envases' + __construct(): var + buscarId(var): var + buscarTipo(var): var + insertar(var): var + listAll(): var + listar(var, var, var): var + modificar(var): var	# table: var = 'envxcer' + __construct(): var + buscarCerEnv(var, var): var + buscarId(var): var + contarXCerveza(var): var + insertar(var): var + insertarGetId(var): var + listAll(): var + listar(var, var, var): var + listarXCerveza(var): var + modificar(var): var
SingletonAbstractDAO DBLineaPedidoDAO	SingletonAbstractDAO DBPedidoDAO	SingletonAbstractDAO DBRolDAO	SingletonAbstractDAO DBSucursalDAO	SingletonAbstractDAO DBUsuarioDAO
# table: var = 'lineapedido' + __construct(): var + buscarId(var): var + insertar(var): var + listar(var, var, var): var + modificar(var): var	# table: var = 'pedidos' + __construct(): var + buscarId(var): var + insertar(var): var + listar(var, var, var): var + modificar(var): var	# table: var = 'roles' + __construct(): var + buscarId(var): var + buscarRol(var): var + insertar(var): var + listAll(): var + listar(var, var, var): var + modificar(var): var	# table: var = 'sucursales' + __construct(): var + buscarId(var): var + buscarNombre(var): var + insertar(var): var + listAll(): var + listar(var, var, var): var + modificar(var): var	# table: var = 'usuarios' + __construct(): var + buscarEmail(var): var + buscarId(var): var + contarUsuario(): var + insertar(var): var + listar(var, var, var): var + modificar(var): var

Figure 5: DB

Connection

Class in package 'DB'

Connection
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

OPERATIONS
◆ Connect () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBCervezaDAO

Class in package 'DB'

DBCervezaDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS
↳ Realization from DBCervezaDAO to iDao [Direction is 'Source -> Destination'.]

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from DBCervezaDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
ATTRIBUTES	
◆ table : var Protected = 'cervezas'	[Is static False. Containment is Not Specified.]
OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarTipo (tipo : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ contarActivos () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listAll () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listar (page : var, campo : var, orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listarConEnv (page : var, campo : var, orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listarConEnv1 (page : var, campo : var, orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listarConEnv2 (page : var, campo : var, orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listarSinStandBy (page : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listarSinStandByLimit () : var Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ modificar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ productoConEnv (id : var, page : var, campo : var, orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBClienteDAO

Class in package 'DB'

DBClienteDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from DBClienteDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
↳ Realization from DBClienteDAO to iDao	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ table : var Protected = 'usuarios'	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarEmail (email : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ contarCliente () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ listar (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ modificar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBEnvaseDAO

Class in package 'DB'

DBEnvaseDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from DBEnvaseDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
↳ Realization from DBEnvaseDAO to iDao	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ table : var Protected = 'envases'	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarTipo (tipo : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listAll () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listar (page : var , campo : var , orden : var) : var Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ modificar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBEnvxcerDAO

Class in package 'DB'

DBEnvxcerDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Realization from DBEnvxcerDAO to iDao	[Direction is 'Source -> Destination'.]
↳ Generalization from DBEnvxcerDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ table : var Protected = 'envxcer'	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarCerEnv (idCer : var, idEnv : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ contarXCerveza (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertarGetId (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ listAll () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listar (page : var , campo : var , orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listarXCerveza (idCerveza : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ modificar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBLineaPedidoDAO

Class in package 'DB'

DBLineaPedidoDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Realization from DBLineaPedidoDAO to iDao	[Direction is 'Source -> Destination'.]
↳ Generalization from DBLineaPedidoDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ table : var Protected = 'lineapedido'	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listar (page : var , campo : var , orden : var) : var Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ modificar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBPedidoDAO

Class in package 'DB'

DBPedidoDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from DBPedidoDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
↳ Realization from DBPedidoDAO to iDao	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ table : var Protected = 'pedidos'	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ listar (page : var, campo : var, orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ modificar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBRoleDAO

Class in package 'DB'

DBRoleDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from DBRoleDAO to SingletonAbstractDAO

[Direction is 'Source -> Destination'.]

↳ Realization from DBRoleDAO to iDao

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ table : var Protected = 'roles'

[Is static False. Containment is Not Specified.]

OPERATIONS

◆ __construct () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ buscarId (id : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ buscarRol (rol : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ insertar (obj : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ listAll () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ listar (page : var, campo : var, orden : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ modificar (obj : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBSucursalDAO

Class in package 'DB'

DBSucursalDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from DBSucursalDAO to SingletonAbstractDAO

[Direction is 'Source -> Destination'.]

↳ Realization from DBSucursalDAO to iDao

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ table : var Protected = 'sucursales'

[Is static False. Containment is Not Specified.]

OPERATIONS

◆ __construct () : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ buscarId (id : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ buscarNombre (nombre : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ insertar (obj : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ listar (page : var, campo : var, orden : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ modificar (obj : var) : var Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

DBUsuarioDAO

Class in package 'DB'

DBUsuarioDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends SingletonAbstractDAO

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Realization from DBUsuarioDAO to iDao	[Direction is 'Source -> Destination'.]
↳ Generalization from DBUsuarioDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
⌚ table : var Protected = 'usuarios'	[Is static False. Containment is Not Specified.]

OPERATIONS	
addCriterion () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
buscarEmail (email : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
contarUsuario () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
listar (page : var, campo : var, orden : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
modificar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

SingletonAbstractDAO

Class in package 'Daos'

SingletonAbstractDAO
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

INCOMING STRUCTURAL RELATIONSHIPS	
-----------------------------------	--

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from DBSucursalDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from Request to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBRolDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBEnvaseDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBLineaPedidoDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBPedidoDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBClienteDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBCervezaDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBUsuarioDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]
⇒ Generalization from DBEnvxcerDAO to SingletonAbstractDAO	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ instancia : var Private = array()	[Is static True. Containment is Not Specified.]

OPERATIONS	
◆ getInstance () : var Public	[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

iDao
 Version 1.0 Phase 1.0 Proposed
 Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Realization from DBLineaPedidoDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBUsuarioDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBEnvaseDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBCervezaDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBSucursalDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBRolDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBClienteDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBPedidoDAO to iDao	[Direction is 'Source -> Destination'.]
⇒ Realization from DBEnvxcerDAO to iDao	[Direction is 'Source -> Destination'.]

OPERATIONS	
◆ buscarId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ contar () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ eliminar (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ insertar (obj : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
◆ listar (page : var , campo : var , orden : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ modificar (obj : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ modificarStandBy (id : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Models

Package in package 'Domain Model'



Models diagram

Class diagram in package 'Models'

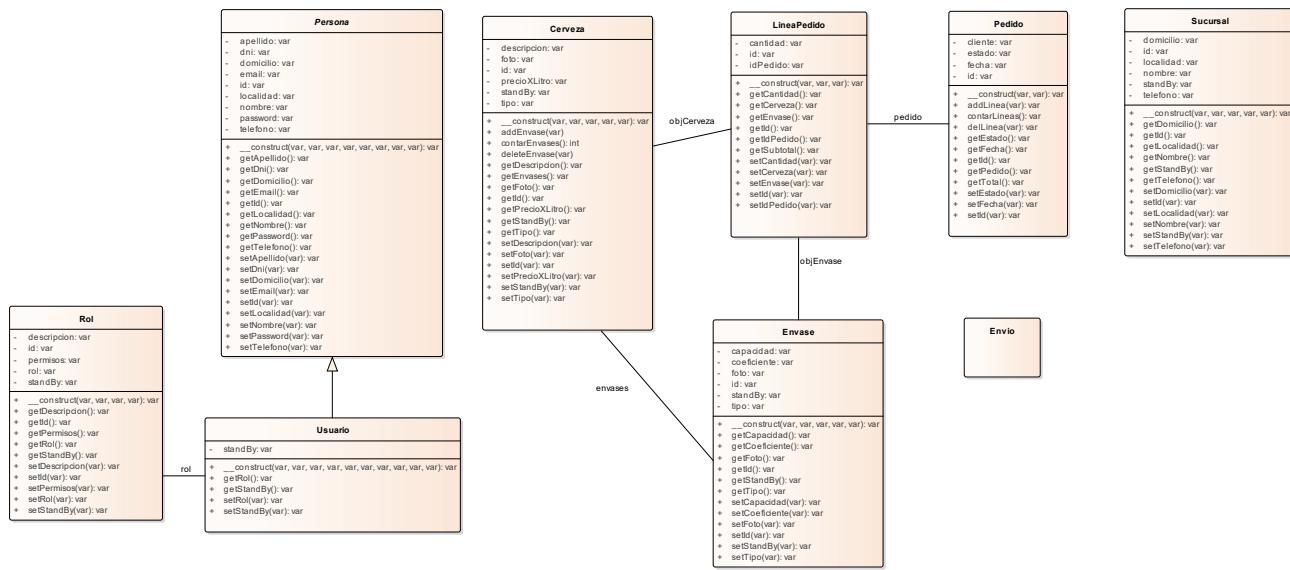


Figure 6: Models

Cerveza

Class in package 'Models'



ATTRIBUTES	
◆ descripcion : var	Private [Is static False. Containment is Not Specified.]
◆ foto : var	Private [Is static False. Containment is Not Specified.]
◆ id : var	Private [Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ precioXLitro : var Private	[Is static False. Containment is Not Specified.]
◆ standBy : var Private	[Is static False. Containment is Not Specified.]
◆ tipo : var Private	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✓ . Association (direction: Unspecified) envases	
Source: Public (Class) Cerveza	Target: Public (Class) Envase
✓ . Association (direction: Unspecified) objCerveza	
Source: Public (Class) LineaPedido	Target: Public (Class) Cerveza

OPERATIONS	
◆ __construct (tipo : var , descripcion : var , precioXLitro : var , foto : var , standBy : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addEnvase (envase : var) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ contarEnvases () : int Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ deleteEnvase (envase : var) : Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getDescripcion () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getEnvases () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getFoto () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getId () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆	getPrecioXLitro () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	getStandBy () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	getTipo () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setDescripcion (descripcion : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setFoto (foto : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setId (id : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setPrecioXLitro (precioXLitro : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setStandBy (standBy : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setTipo (tipo : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Cliente

Class in package 'Models'

Cliente
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends Persona

OUTGOING STRUCTURAL RELATIONSHIPS	
↳	Generalization from Cliente to Persona
[Direction is 'Source -> Destination'.]	
OPERATIONS	
◆	__construct (nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

Envase

Class in package 'Models'

Envase
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES

◆ capacidad : var Private
[Is static False. Containment is Not Specified.]

◆ coeficiente : var Private
[Is static False. Containment is Not Specified.]

◆ foto : var Private
[Is static False. Containment is Not Specified.]

◆ id : var Private
[Is static False. Containment is Not Specified.]

◆ standBy : var Private
[Is static False. Containment is Not Specified.]

◆ tipo : var Private
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Unspecified) envases

Source: Public (Class) Cerveza

Target: Public (Class) Envase

↙ Association (direction: Unspecified) objEnvase

Source: Public (Class) LineaPedido

Target: Public (Class) Envase

OPERATIONS

◆ __construct (tipo : var , capacidad : var , coeficiente : var , foto : var , standBy : var) : var Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getCapacidad () : var Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆ getCoeficiente () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getFoto () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getId () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getStandBy () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTipo () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setCapacidad (capacidad : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setCoeficiente (coeficiente : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setFoto (foto : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setStandBy (standBy : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setTipo (tipo : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Envio

Class in package 'Models'

Envio
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017

Envio

Class in package 'Models'

Envio
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 23/11/2017. Last modified 23/11/2017

LineaPedido*Class in package 'Models'*

LineaPedido
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES

- ◆ cantidad : var Private
[Is static False. Containment is Not Specified.]
- ◆ id : var Private
[Is static False. Containment is Not Specified.]
- ◆ idPedido : var Private
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

- ↙ Association (direction: Unspecified) objCerveza
Source: Public (Class) LineaPedido Target: Public (Class) Cerveza
- ↙ Association (direction: Unspecified) objEnvase
Source: Public (Class) LineaPedido Target: Public (Class) Envase
- ↙ Association (direction: Unspecified) pedido
Lista de lineas de pedido
Source: Public (Class) Pedido Target: Public (Class) LineaPedido

OPERATIONS

- ◆ __construct (objCerveza : var , objEnvase : var , cantidad : var) : var Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ getCantidad () : var Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
- ◆ getCerveza () : var Public

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ getEnvase () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getId () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getIdPedido () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getSubtotal () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setCantidad (cantidad : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setCerveza (objCerveza : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setEnvase (objEnvase : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setIdPedido (idPedido : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Pedido

Class in package 'Models'

Pedido
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◆ cliente : var Private	[Is static False. Containment is Not Specified.]
◆ estado : var Private	
3 estados 'En Preparacion' - 'Enviado' - 'Entregado'	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ fecha : var Private	[Is static False. Containment is Not Specified.]
◆ id : var Private	[Is static False. Containment is Not Specified.]
ASSOCIATIONS	
<p>✓ . Association (direction: Unspecified) pedido</p> <p>Lista de lineas de pedido</p> <p>Source: Public (Class) Pedido Target: Public (Class) LineaPedido</p>	
OPERATIONS	
◆ __construct (fecha : var , estado : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ addLinea (linea : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ contarLineas () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ delLinea (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getEstado () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getFecha () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getId () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPedido () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTotal () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setEstado (estado : var) : var Public	

OPERATIONS	
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]	
◆ setFecha (fecha : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setId (id : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Persona

Class in package 'Models'

Persona
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from Cliente to Persona	[Direction is 'Source -> Destination'.]
⇒ Generalization from Usuario to Persona	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ apellido : var Private	[Is static False. Containment is Not Specified.]
◆ dni : var Private	[Is static False. Containment is Not Specified.]
◆ domicilio : var Private	[Is static False. Containment is Not Specified.]
◆ email : var Private	[Is static False. Containment is Not Specified.]
◆ id : var Private	[Is static False. Containment is Not Specified.]
◆ localidad : var Private	[Is static False. Containment is Not Specified.]
◆ nombre : var Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ password : var	Private [Is static False. Containment is Not Specified.]
◆ telefono : var	Private [Is static False. Containment is Not Specified.]
OPERATIONS	
◆ __construct	(nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getApellido ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getDni ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getDomicilio ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getEmail ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getId ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getLocalidad ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getNombre ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getPassword ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getTelefono ()	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setApellido (apellido : var)	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setDni (dni : var)	: var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
◆ setDomicilio (domicilio : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setEmail (email : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setId (id : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setLocalidad (localidad : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setNombre (nombre : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setPassword (password : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setTelefono (telefono : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Rol

Class in package 'Models'

Rol
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES
◆ descripcion : var Private [Is static False. Containment is Not Specified.]
◆ id : var Private [Is static False. Containment is Not Specified.]
◆ permisos : var Private [Is static False. Containment is Not Specified.]
◆ rol : var Private [Is static False. Containment is Not Specified.]
◆ standBy : var Private

ATTRIBUTES
[Is static False. Containment is Not Specified.]
ASSOCIATIONS
<p> Association (direction: Unspecified) rol</p> <p>Source: Public (Class) Usuario Target: Public (Class) Rol</p>
OPERATIONS
<p> __construct (rol : var , descripcion : var , permisos : var , standBy : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> getDescripcion () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> getId () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> getPermisos () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> getRol () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> getStandBy () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> setDescripcion (descripcion : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> setId (id : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> setPermisos (permisos : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> setRol (rol : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<p> setStandBy (standBy : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>

Sucursal

Class in package 'Models'

Sucursal
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES	
◆ domicilio : var Private	[Is static False. Containment is Not Specified.]
◆ id : var Private	[Is static False. Containment is Not Specified.]
◆ localidad : var Private	[Is static False. Containment is Not Specified.]
◆ nombre : var Private	[Is static False. Containment is Not Specified.]
◆ standBy : var Private	[Is static False. Containment is Not Specified.]
◆ telefono : var Private	[Is static False. Containment is Not Specified.]

OPERATIONS	
◆ __construct (nombre : var , domicilio : var , localidad : var , telefono : var , standBy : var) : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getDomicilio () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getId () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getLocalidad () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getNombre () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getStandBy () : var Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS	
◆	getTelefono () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setDomicilio (domicilio : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setId (id : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setLocalidad (localidad : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setNombre (nombre : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setStandBy (standBy : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆	setTelefono (telefono : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Usuario

Class in package 'Models'

Usuario
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017
Extends Persona

OUTGOING STRUCTURAL RELATIONSHIPS	
↳	Generalization from Usuario to Persona [Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆	standBy : var Private [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
↙	Association (direction: Unspecified) rol Source: Public (Class) Usuario Target: Public (Class) Rol

OPERATIONS
◆ __construct (rol : var , nombre : var , apellido : var , domicilio : var , localidad : var , telefono : var , dni : var , email : var , password : var , standBy : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getRol () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getStandBy () : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setRol (rol : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setStandBy (standBy : var) : var Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Utils

Package in package 'Domain Model'

Utils
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

Utils diagram

Class diagram in package 'Utils'

Utils
Version 1.0
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

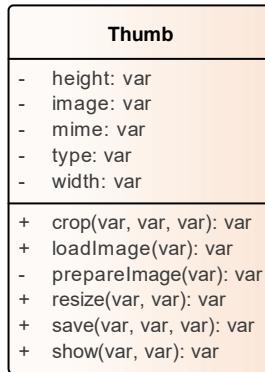


Figure 7: Utils

Thumb

Class in package 'Utils'

Thumb
Version 1.0 Phase 1.0 Proposed
Rodrigo Soria created on 21/11/2017. Last modified 21/11/2017

ATTRIBUTES
◆ height : var Private [Is static False. Containment is Not Specified.]
◆ image : var Private [Is static False. Containment is Not Specified.]
◆ mime : var Private [Is static False. Containment is Not Specified.]
◆ type : var Private [Is static False. Containment is Not Specified.]

ATTRIBUTES
◆ width : var Private [Is static False. Containment is Not Specified.]
OPERATIONS
◆ crop (cwidth : var , cheight : var , pos : var) : var Public ---MÁtodo de extraer una sección de la imagen sin deformarla [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ loadImage (name : var) : var Public ---MÁtodo de leer la imagen [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ prepareImage (image : var) : var Private ---MÁtodo privado de tratar las imágenes antes de mostrarlas [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ resize (value : var , prop : var) : var Public ---MÁtodo de redimensionar la imagen sin deformarla [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ save (name : var , quality : var , type : var) : var Public ---MÁtodo de guardar la imagen [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ show (type : var , base64 : var) : var Public ---MÁtodo de mostrar la imagen sin guardarla [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]