

FIND A PLAN

BY ABELIAN GROUP

SRS: Especificación de requisitos de software.

Víctor Carrillo Redondo

Óscar Díaz Ribagorda

Mateo García Pérez

Pablo Hidalgo Palencia

Daniel Loscos Barroso

Javier Pastor Ramírez

Lucas de Torre Barrio

CONTROL DE CAMBIOS

Número de versión	Fecha	Autores	Descripción
Versión 0	10/12/2016		Apertura de la plantilla para la SRS.
Versión 1	10/12/2016	Pablo Hidalgo	Completado el primer apartado de la SRS con la información de la empresa y la aplicación.
Versión 2	10/12/2016	Oscar Díaz	Completado segunda parte de la SRS con las características de la aplicación.
Versión 3	10/12/2016	Pablo Hidalgo, Javier Pastor	Incluir en la SRS el tercer apartado con las especificaciones sobre la interfaz y las funciones de la aplicación.
Versión 4	11 /12/2016	Víctor Carillo, Oscar Díaz, Mateo García, Pablo Hidalgo, Daniel Loscos, Javier Pastor, Lucas de Torre.	Se añaden los casos de uso al apartado 3.2.
Versión 5	30/12/2016	Oscar Díaz	Se ha reorganizado el apartado de funciones (2.2) por subsistemas y se han añadido los casos de uso de Ofertas y Sorteos al apartado 3.2.
Versión 6	30/12/2016	Pablo Hidalgo y Lucas de Torre (Pantallas), Oscar Díaz (Inclusión en el documento).	Se han incluido las plantillas de las pantallas y los diagramas de casos de uso de cada subsistema.
Versión 7	17/1/2016	Oscar Díaz	Se ha cambiado el diseño del SRS para hacerlo más llamativo así como esquematizado y vinculado apartados para la fácil lectura. También se han incluido las CRC's

Índice

Introducción	5
Propósito	5
Alcance	5
Definiciones, acrónimos y abreviaturas	5
Referencias	5
Organización del documento	5
Descripción general	6
Perspectiva del producto	6
Funciones del producto	7
Ofertas.	7
Particulares	7
Empresas	7
Social	7
Buscador	8
Gestión	8
Particulares	8
Empresas	8
Sorteos	9
Particulares	9
Empresas	9
Cupones	9
Características del usuario	9
Restricciones	10
Supuestos y dependencias	10
Requisitos futuros	10
Requisitos específicos	12
Interfaces externos	12
Particulares	12
Empresas	12
Extensión de Verificación	12
Requisitos funcionales (Casos de uso)	14
Buscador	14
Diagrama de casos de uso	14
Casos de uso	15
Caso de uso: Buscar Oferta	15
Caso de uso: Filtrar Búsqueda Oferta.	16
Caso de uso: Ordenar Búsqueda oferta.	17

Caso de uso: Buscar Sorteo	18
Caso de uso: Filtrar Búsqueda Sorteo.	19
Caso de uso: Ordenar Búsqueda Sorteo.	20
Cupones	21
Diagrama de casos de uso	21
Casos de uso	21
Caso de uso: Consultar Cupones.	21
Caso de uso: Verificación.	23
Caso de uso: Modificar Cupones.	24
Gestión	25
Diagrama de casos de uso	25
Casos de uso	25
Caso de uso: Administrar Cuenta (Empresas).	25
Caso de uso: Comprar espacios	27
Caso de uso: Confirmación de registros.	28
Caso de uso: Registro Empresa.	29
Caso de uso: Registro Usuario	30
Caso de uso: Login Usuario	32
Caso de uso: Editar Avatar	33
Caso de uso: Mostrar/Ocultar Insignias	34
Caso de uso: Logout	35
Social	36
Diagrama de casos de uso	36
Casos de uso	36
Caso de uso: Poner Comentario.	36
Caso de uso: Compartir con Amigos.	38
Caso de uso: Apuntarse al Plan.	39
Ofertas	40
Diagrama de casos de uso	40
Casos de uso	40
Caso de uso: Crear Oferta.	40
Caso de uso: Editar Oferta.	42
Caso de uso: Eliminar oferta (Empresa).	43
Caso de uso: Publicar oferta.	44
Caso de uso: Consultar ofertas.	45
Caso de uso: Eliminar oferta (Admin).	46
Caso de uso: Aceptar Oferta	47
Caso de uso: Generar QR (Oferta)	48
Sorteos	49
Diagrama de casos de uso	49
Casos de uso	49
Caso de uso: Crear Sorteo.	49
Caso de uso: Editar Sorteo.	51
Caso de uso: Eliminar sorteo (Empresa).	52
Caso de uso: Publicar sorteo.	53
Caso de uso: Sortear	54
Caso de uso: Comprobar Ganador	55
Caso de uso: Consultar sorteos.	56
Caso de uso: Eliminar sorteo (Admin).	57
Caso de uso: Participar	58

Caso de uso: Generar QR (Sorteo)	59
Requisitos Funcionales (CRC`s)	60
Ofertas	61
Sorteos	63
Buscador	65
Gestión	66
Social	77
Cupones	80
Requisitos de rendimiento	80
Requisitos lógicos de la base de datos	80
Restricciones de diseño	80
Atributos del sistema software	81
<i>Apéndices</i>	82

INTRODUCCIÓN

PROPÓSITO

En este documento explicaremos detalladamente el funcionamiento de una aplicación destinada tanto a empresas para que propongan planes de ocio, como a cualquier público interesado para que los consuma e interaccione con otros usuarios, principalmente.

ALCANCE

La aplicación se llamará *Find a Plan* y sus funcionalidades serán: permitir a las empresas publicar planes de ocio, que pueden asimismo llevar descuentos asociados por utilizar la aplicación y sorteos (de diferentes tipos) ofreciendo productos o servicios suyos; pero la aplicación no permite comprar directamente productos de la empresa.

Los usuarios podrán apuntarse a estos planes y sorteos, y compartirlos con sus amigos o también con desconocidos que también estén interesados, con el objetivo de poder conocer a gente con los mismos gustos.

Además, cuantos más planes se publiquen o consuman, más relevancia y ventajas tendrá la empresa o el particular dentro de la aplicación (esto incentivará el uso de la aplicación).

Como vemos, la aplicación utiliza y amplía los beneficios de otras aplicaciones del sector de ocio ya existentes.

DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

En principio no se utilizarán muchas abreviaturas o términos novedosos en estas páginas.

Podríamos destacar que el nombre de la aplicación se podrá ver abreviado a FAP, como ejemplo.

REFERENCIAS

Como referencias para el desarrollo de este proyecto hemos tomado otras empresas como Fever y Atrápalo que tienen como objetivo la oferta de planes de ocio como nosotros. Para el desarrollo de una aplicación innovadora y moderna tenemos como referencia a aplicaciones punteras como Twitter o Facebook de las cuales sacamos su aspecto social.

ORGANIZACIÓN DEL DOCUMENTO

En el segundo apartado de este documento SRS pasaremos a explicar más detenidamente las características generales de la aplicación y ciertos factores que influirán en el desarrollo, como los requisitos o las restricciones. En la tercera sección desglosaremos mucho más todas las funciones, requisitos y casos de uso para que quede todo preparado para que un desarrollador pueda trabajar sobre los requisitos que se explican.

DESCRIPCIÓN GENERAL

PERSPECTIVA DEL PRODUCTO

La aplicación FAP se trata de un producto autónomo que se utilizará desde el móvil y para el cual tendremos que construir una base de datos para almacenar la información.

Los usuarios que utilizan la aplicación son:



FUNCIONES DEL PRODUCTO

Los subsistemas en los que dividimos las funciones son los siguientes (tienen vínculos para poder acceder rápidamente a los diferentes apartados):



OFERTAS.

PARTICULARES

Guardar Oferta: Si al usuario le interesa la oferta la selecciona para guardar en una lista de ofertas guardadas.

Descargar Cupón de la Oferta: Dada una oferta, se selecciona el botón de descargar cupón para descargar un código QR propio de la oferta.

Quitar Oferta de la Lista de Guardadas: Dada una oferta almacenada en la lista de ofertas guardadas, se selecciona y elimina de la lista.

EMPRESAS

Crear Oferta: Seleccionas la opción de crear oferta e introduces los datos necesarios para completar la plantilla. La oferta se te guarda en una lista de ofertas de la empresa.

Editar Oferta: Dada una oferta ya creada pero no publicada, seleccionas la opción de editar la información de la oferta y cambias los campos deseados de la oferta.

Publicar Oferta: Dada una oferta creada, seleccionas la opción de publicar para que la oferta se muestre a todos los usuarios particulares, esta opción requiere el gasto de 1 token.

Eliminar Oferta: Dada una oferta creada, seleccionas la opción de eliminar oferta para quitarla de la lista de ofertas de la empresa y en caso de estar publicada, dejará de estarlo y se eliminará.

SOCIAL

Realizar Comentario en la Oferta: Dada una oferta, seleccionas la opción de añadir comentario para poder dejar un comentario sobre el campo de comentarios de la oferta.

Realizar Propuesta de Quedada (con Desconocidos): En un apartado de comunicación dentro de cada oferta, los usuarios pueden dejar un mensaje proponiendo una quedada para realizar el plan con el resto de usuarios de la aplicación.

Compartir Oferta con Conocidos: Seleccionas la opción de compartir para poder mandar el enlace a la oferta a tus amigos a través de otras aplicaciones como Facebook, Twitter, etc.

BUSCADOR

Buscar Oferta: Busca una oferta en la lista de ofertas.

Filtrar Oferta: Filtra la búsqueda para únicamente mostrar las ofertas con los parámetros buscados.

Ordenar Ofertas: Ordena las ofertas dado un parámetro (ej. Más populares).

Buscar Sorteo: Busca un sorteo en la lista de sorteos.

Filtrar Sorteos: Filtra la búsqueda para únicamente mostrar los sorteos con los parámetros buscados.

Ordenar Sorteos: Ordena los sorteos dado un parámetro (ej. Más populares).

GESTIÓN

PARTICULARES

Registrarse como Particular: Introduces los datos solicitados para crear una nueva cuenta de particular en la aplicación.

Darse de Baja: Dentro de una cuenta de particular registrada en la aplicación seleccionas la opción de eliminar cuenta para eliminar esa cuenta de la aplicación.

Hacer Log In: Introduces los datos solicitados (correo y contraseña) para poder entrar en la aplicación.

Hacer Log Out: Dada la cuenta que esta con la sesión iniciada, se selecciona la opción de Log Out para salir de la aplicación.

Editar Perfil: Seleccionas la opción de editar perfil para cambiar los campos con la información de tu perfil.

Editar Avatar: Seleccionas la opción de editar avatar para poder cambiar la apariencia de tu avatar con objetos predeterminados, objetos desbloqueados, etc.

EMPRESAS

Registrarse como Empresa: Introduces los datos necesarios para dar de alta una nueva cuenta de empresa en la aplicación.

Darse de Baja: Una cuenta de empresa registrada en la aplicación selecciona la opción de darse de baja para eliminar la cuenta de la aplicación.

Hacer Log In: Introduces los datos solicitados (correo y contraseña) para poder entrar en la aplicación.

Hacer Log Out: Dada la cuenta que esta con la sesión iniciada, se selecciona la opción de Log Out para salir de la aplicación.

Editar Perfil: Seleccionas la opción de editar perfil de empresa para cambiar los campos con la información de la empresa.

Comprar Tokens: Seleccionas la opción de comprar Tokens para poder almacenar más Tokens en tu cuenta con los cuales poder publicar ofertas y sorteos.

SORTEOS

PARTICULARES

Apuntarte a Sorteo: Dado un sorteo, lo seleccionas y pulsas el botón de apuntarte al sorteo para que se guarde tu participación en el sorteo.

EMPRESAS

Crear Sorteo: Seleccionas la opción de crear sorteo y rellenas los campos solicitados para la creación del sorteo, una vez creado se guarda en la lista de la empresa.

Publicar Sorteo: Dado un sorteo creado, puedes seleccionar la opción de publicar para mostrar el sorteo a todos los usuarios particulares.

CUPONES

Verificación de Cupones: Seleccionas la opción de verificar cupón y se utiliza la cámara para registrar el código QR de un cupón perteneciente a algún particular y así registrar que ese particular ha utilizado esa oferta.

CARACTERÍSTICAS DEL USUARIO

La aplicación va dirigida tanto a Particulares como a Empresas.

Para los Particulares se ofrece muchas ofertas y descuentos en planes que pueden utilizar para su disfrute y además así aumentar su nivel en la aplicación (así accede a más sorteos y más cosas). Necesita un mínimo conocimiento en el uso de aplicaciones para móvil (saber descargar la aplicación y registrarse).

Para las Empresas se ofrece publicar ofertas y sorteos para publicitar su marca. Para ello solo necesitan estar familiarizados mínimamente en el uso de aplicaciones móviles y la creación de perfiles (porque de manera muy parecida se crearan las ofertas y sorteos).

RESTRICCIONES

Aplicación desarrollada para Android e iOS.



Necesaria la aplicación en todos los establecimientos.



Base de datos sólida con back up de todo.



Para la rápida transmisión y fácil almacenamiento.

La información guardada ha de ser pequeña.

La interfaz de la aplicación va a ser intuitiva y llamativa.



Particulares y empresas solo verán la parte de la aplicación dirigida a ellos.

Diseño modular para la fácil comprensión del código y la prevención de errores.

SUPUESTOS Y DEPENDENCIAS

Necesitamos que nuestros usuarios tengan acceso a internet.

100% Real

Necesitamos la colaboración de Empresas, sin ellas no tenemos ofertas ni sorteos que ofrecer

Sobre todo, necesitamos que las empresas, ofertas y sorteos que se publiquen sean reales.

REQUISITOS FUTUROS

Se va a empezar la aplicación por las funciones principales que son las que tienen que ver con los perfiles de empresa y particular y los módulos de ofertas y sorteos. Más adelante se irá ampliando con

otras funciones como un ránking de empresas, unos avatares para los usuarios, etc. Para ello necesitaremos que los módulos estén bien distinguidos y que estas nuevas funciones no cambien a las ya implementadas. Las nuevas funcionalidades no desviarán del objetivo principal que son las ofertas de las empresas, solo le añadirán funcionalidad (como por ejemplo una opción para comentar en las ofertas).

REQUISITOS ESPECÍFICOS

INTERFACES EXTERNOS

Las interfaces que precisará la aplicación serán en principio tres (todas en plataforma móvil): la de los usuarios, la de las empresas y una extensión que permitirán verificar que un usuario ha participado en el plan en el que se apuntó. Hemos de crear esta nueva extensión porque no es lo mismo que la interfaz de las empresas. Esto lo podemos ver porque la situación habitual será que los community manager sean quienes manejen la aplicación por parte de la empresa para subir ofertas o sorteos, pero serán por ejemplo los dependientes quienes verifiquen los clientes que van. Como se trata de diferentes trabajadores, necesitamos separar funcionalidades también en la aplicación.



PARTICULARES

EMPRESAS

EXTENSIÓN DE VERIFICACIÓN

Es importante saber si un usuario finalmente consume una oferta de ocio, tras haber hecho el plan a través de la aplicación. Para esta verificación, se precisará de un sistema generador de códigos QR específicos de cada plan, que los usuarios podrán descargarse cuando se apunten a un plan. Cuando los usuarios vayan a este plan, la empresa necesitará una extensión que lea estos códigos y mande la información necesaria al sistema para que quede registrada esa visita y, por lo tanto, se aumenten los puntos del usuario y de la empresa en cuestión.

Por lo tanto, esta extensión será en esencia funcional, dispondrá de un lector y una interfaz sencilla que, en tan solo un par de clicks, permita que se suba la información al sistema.

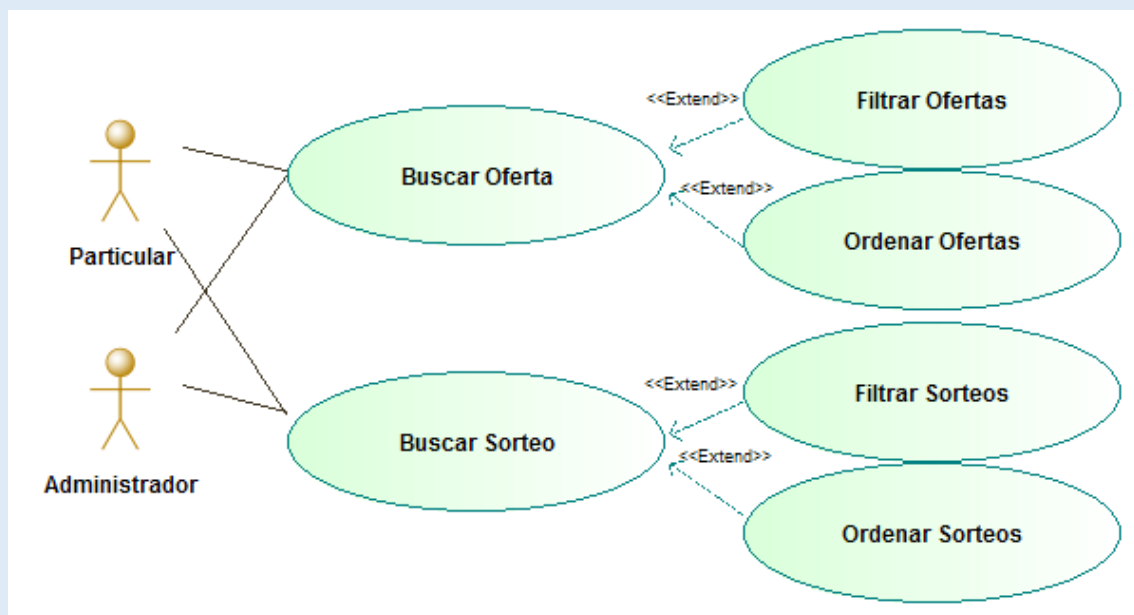
REQUISITOS FUNCIONALES (CASOS DE USO)

En este apartado se muestran todos los casos de uso distribuidos por los diferentes subsistemas: Buscador, Cupones, Gestión, Social, Ofertas y Sorteos.



BUSCADOR

DIAGRAMA DE CASOS DE USO



CASOS DE USO

CASO DE USO: BUSCAR OFERTA

Nombre del AUTOR: Víctor Carrillo Redondo.

Identificador: B01

Objetivo en Contexto:

Buscar en la base de datos de ofertas o de empresas la información introducida por el usuario de la aplicación.

Actor principal: Particular.

Actores secundarios: Base de Datos de Empresas y Base de Datos de Ofertas.

Qué datos usa: Ofertas publicadas y empresas registradas en la aplicación.

Precondiciones:

El PARTICULAR tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: El sistema muestra los resultados obtenidos tras la búsqueda.

Fallo: El sistema da un mensaje de error y vuelve al menú anterior.

Flujo principal

1. El PARTICULAR selecciona la opción de buscar pulsando sobre el icono de la lupa.
2. El SISTEMA le pide los datos sobre los que realizar la búsqueda en las Bases de Datos de Empresas y Ofertas.
3. El PARTICULAR introduce los datos.
4. El SISTEMA muestra los resultados de la búsqueda.

Flujos secundarios

- 1-a. El PARTICULAR puede seleccionar la opción de "Búsqueda Avanzada" si quiere ordenar los resultados obtenidos o si quiere filtrar la búsqueda.
- 3-a. Si se produce algún error durante la introducción de los datos, el SISTEMA muestra un mensaje de error y vuelve al menú anterior.
- 4-a. Si no se encontró nada relacionado con los datos introducidos por el PARTICULAR, el SISTEMA muestra un mensaje de error avisando de que no se encontró nada.

CASO DE USO: FILTRAR BÚSQUEDA OFERTA.

Nombre del AUTOR: Víctor Carrillo Redondo.

Identificador: B02

Objetivo en Contexto:

Al hacer una búsqueda, el SISTEMA solo muestra aquellos resultados que cumplen las condiciones del filtro seleccionado por el PARTICULAR.

Actor principal: Particular.

Actores secundarios: Bases de Datos de Empresas y Ofertas.

Qué datos usa: Ofertas publicadas y empresas registradas en la aplicación.

Precondiciones:

El PARTICULAR tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: El sistema muestra los resultados obtenidos tras la búsqueda.

Fallo: El sistema da un mensaje de error y vuelve al menú anterior.

Flujo principal

1. El PARTICULAR selecciona la opción de buscar pulsando sobre el icono de la lupa, y selecciona la opción de "Búsqueda Avanzada".
2. El SISTEMA le pide los datos sobre los que realizar la búsqueda en las Bases de Datos de Empresas y Ofertas, además del filtro que quiere aplicar a los resultados.
3. El PARTICULAR introduce los datos y selecciona el filtro que quiere que pasen los resultados de la búsqueda (por ejemplo: por precio, empresa que hizo la oferta, ect).
4. El SISTEMA muestra los resultados de la búsqueda filtrada.

Flujos secundarios

- 3-a. Si se produce algún error durante la introducción de los datos, el SISTEMA muestra un mensaje de error y vuelve al menú anterior.
- 4-a. Si no se encontró nada relacionado con los datos introducidos por el PARTICULAR, el SISTEMA muestra un mensaje de error avisando de que no se encontró nada.

CASO DE USO: ORDENAR BÚSQUEDA OFERTA.

Nombre del AUTOR: Víctor Carrillo Redondo.

Identificador: B03.

Objetivo en Contexto:

Al hacer una búsqueda, el SISTEMA ordena los resultados obtenidos según el criterio elegido por el usuario, por ejemplo según la fecha de vencimiento de la oferta, o según el precio.

Actor principal: Particular.

Actores secundarios: Bases de Datos de Empresas y Ofertas.

Qué datos usa: Ofertas publicadas y empresas registradas en la aplicación.

Precondiciones:

El PARTICULAR debe tener la sesión iniciada.

Postcondiciones:

Éxito: El SISTEMA muestra los resultados obtenidos tras la búsqueda de manera ordenada según eligió el PARTICULAR.

Fallo: El sistema da un mensaje de error y vuelve al menú anterior.

Flujo principal

1. El PARTICULAR selecciona la opción de buscar pulsando sobre el icono de la lupa, y selecciona la opción de "Búsqueda Avanzada".
2. El SISTEMA le pide los datos sobre los que realizar la búsqueda en las Bases de Datos de Empresas y Ofertas, además del criterio de ordenación de los resultados.
3. El PARTICULAR introduce los datos y selecciona el criterio bajo el que quiere que se ordenen los resultados obtenidos.
4. El SISTEMA muestra los resultados de la búsqueda de forma ordenada.

Flujos secundarios

- 3-a. Si se produce algún error durante la introducción de los datos, el SISTEMA muestra un mensaje de error y vuelve al menú anterior.
- 4-a. Si no se encontró nada relacionado con los datos introducidos por el PARTICULAR, el SISTEMA muestra un mensaje de error avisando de que no se encontró nada.

CASO DE USO: BUSCAR SORTEO

Nombre del AUTOR: Daniel Loscos Barroso.

Identificador: B11

Objetivo en Contexto:

Buscar en la base de datos de sorteos o de empresas la información introducida por el usuario de la aplicación.

Actor principal: Particular.

Actores secundarios: Base de Datos de Empresas y Base de Datos de Sorteos.

Qué datos usa: Sorteos publicados y empresas registradas en la aplicación.

Precondiciones:

El PARTICULAR tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: El sistema muestra los resultados obtenidos tras la búsqueda.

Fallo: El sistema da un mensaje de error y vuelve al menú anterior.

Flujo principal

1. El PARTICULAR selecciona la opción de buscar pulsando sobre el icono de la lupa.
2. El SISTEMA le pide los datos sobre los que realizar la búsqueda en las Bases de Datos de Empresas y Sorteos.
3. El PARTICULAR introduce los datos.
4. El SISTEMA muestra los resultados de la búsqueda.

Flujos secundarios

- 1-a. El PARTICULAR puede seleccionar la opción de "Búsqueda Avanzada" si quiere ordenar los resultados obtenidos o si quiere filtrar la búsqueda.
- 3-a. Si se produce algún error durante la introducción de los datos, el SISTEMA muestra un mensaje de error y vuelve al menú anterior.
- 4-a. Si no se encontró nada relacionado con los datos introducidos por el PARTICULAR, el SISTEMA muestra un mensaje de error avisando de que no se encontró nada.

CASO DE USO: FILTRAR BÚSQUEDA SORTEO.

Nombre del AUTOR: Daniel Loscos Barroso.

Identificador: B12

Objetivo en Contexto:

Al hacer una búsqueda, el SISTEMA solo muestra aquellos resultados que cumplen las condiciones del filtro seleccionado por el PARTICULAR.

Actor principal: Particular.

Actores secundarios: Bases de Datos de Empresas y Sorteos.

Qué datos usa: Sorteos publicadas y empresas registradas en la aplicación.

Precondiciones:

El PARTICULAR tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: El sistema muestra los resultados obtenidos tras la búsqueda.

Fallo: El sistema da un mensaje de error y vuelve al menú anterior.

Flujo principal

1. El PARTICULAR selecciona la opción de buscar pulsando sobre el icono de la lupa, y selecciona la opción de "Búsqueda Avanzada".
2. El SISTEMA le pide los datos sobre los que realizar la búsqueda en las Bases de Datos de Empresas y Sorteos, además del filtro que quiere aplicar a los resultados.
3. El PARTICULAR introduce los datos y selecciona el filtro que quiere que pasen los resultados de la búsqueda (por ejemplo: criterios de entrada, empresa que hizo la oferta, ect).
4. El SISTEMA muestra los resultados de la búsqueda filtrada.

Flujos secundarios

- 3-a. Si se produce algún error durante la introducción de los datos, el SISTEMA muestra un mensaje de error y vuelve al menú anterior.
- 4-a. Si no se encontró nada relacionado con los datos introducidos por el PARTICULAR, el SISTEMA muestra un mensaje de error avisando de que no se encontró nada.

CASO DE USO: ORDENAR BÚSQUEDA SORTEO.

Nombre del AUTOR: Daniel Loscos Barroso.

Identificador: B13.

Objetivo en Contexto:

Al hacer una búsqueda, el SISTEMA ordena los resultados obtenidos según el criterio elegido por el usuario, por ejemplo: según la fecha de vencimiento del sorteo, o según coste de participar.

Actor principal: Particular.

Actores secundarios: Bases de Datos de Empresas y Sorteos.

Qué datos usa: Sorteos publicadas y empresas registradas en la aplicación.

Precondiciones:

El PARTICULAR debe tener la sesión iniciada.

Postcondiciones:

Éxito: El SISTEMA muestra los resultados obtenidos tras la búsqueda de manera ordenada según eligió el PARTICULAR.

Fallo: El sistema da un mensaje de error y vuelve al menú anterior.

Flujo principal

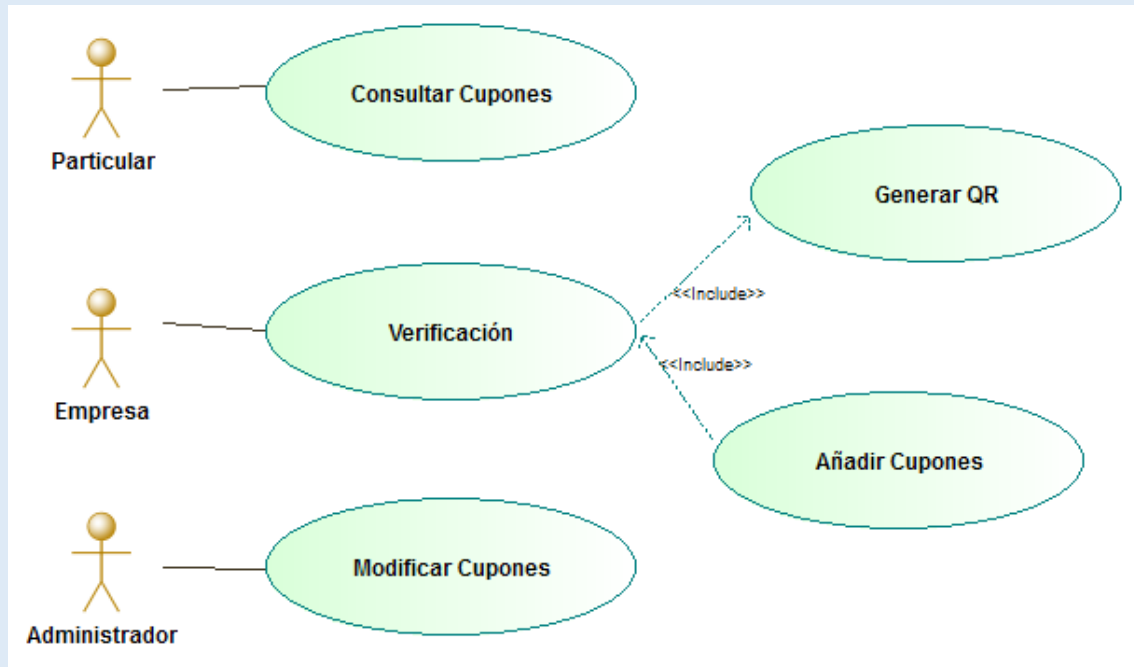
1. El PARTICULAR selecciona la opción de buscar pulsando sobre el icono de la lupa, y selecciona la opción de "Búsqueda Avanzada".
2. El SISTEMA le pide los datos sobre los que realizar la búsqueda en las Bases de Datos de Empresas y Sorteos, además del criterio de ordenación de los resultados.
3. El PARTICULAR introduce los datos y selecciona el criterio bajo el que quiere que se ordenen los resultados obtenidos.
4. El SISTEMA muestra los resultados de la búsqueda de forma ordenada.

Flujos secundarios

- 3-a. Si se produce algún error durante la introducción de los datos, el SISTEMA muestra un mensaje de error y vuelve al menú anterior.
- 4-a. Si no se encontró nada relacionado con los datos introducidos por el PARTICULAR, el SISTEMA muestra un mensaje de error avisando de que no se encontró nada.

CUPONES

DIAGRAMA DE CASOS DE USO



CASOS DE USO

CASO DE USO: CONSULTAR CUPONES.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: C0.

Objetivo en Contexto:

Consulta de cuantos cupones ha acumulado el usuario con cada una de las empresas. Mostrará una lista con todas las empresas que hayan otorgado algún cupón al particular.

Actor principal: Particular.

Actores secundarios: Base de Datos de Usuarios.

Qué datos usa: Datos sobre los cupones que ha acumulado el particular.

Precondiciones:

El particular tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: Muestra una lista de empresas y los cupones acumulados con cada una de ellas.

Fallo: Se muestra un mensaje de error.

Flujo principal

1. El PARTICULAR selecciona la opción del menú que pone “Consultar Cupones”.
2. El SISTEMA solicita a la base de datos, la información sobre cupones acumulados.
3. El SISTEMA muestra la lista de empresas y cupones con cada una del PARTICULAR.

Flujos secundarios

- 3-a. Si el particular no ha acumulado ningún cupón todavía, el SISTEMA muestra el siguiente mensaje: “Aún no tienes ningún cupón, ¡consíguelos participando en ofertas!”.
- 3-b. Si el SISTEMA no recibe correctamente los datos solicitados, el SISTEMA muestra un mensaje de error y lleva al particular al menú anterior.

CASO DE USO: VERIFICACIÓN.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: C11.

Objetivo en Contexto:

Verifica el código QR para confirmar que el PARTICULAR ha completado la oferta.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Usuarios.

Qué datos usa: Datos sobre las ofertas aceptadas por el particular.

Precondiciones:

La Empresa tiene que haber iniciado sesión. El particular tiene que haber aceptado la oferta. Tiene que haberse generado el QR.

Postcondiciones:

Éxito: Muestra un mensaje de éxito e inicia la operación de otorgar puntos al particular.

Fallo: Se muestra un mensaje de error, vuelve al menú anterior.

Flujo principal

1. La EMPRESA selecciona la oferta que va a verificar.
2. La EMPRESA escanea el código QR del móvil del PARTICULAR.
3. El SISTEMA solicita a la base de datos, la información sobre ofertas aceptadas del particular.
4. El SISTEMA comprueba que la oferta seleccionada está en la lista de ofertas aceptadas
5. El SISTEMA confirma el éxito de la operación y comienza el proceso de otorgar los cupones al PARTICULAR.

Flujos secundarios

- 4-a. Si el particular no ha aceptado esa oferta, el SISTEMA muestra un mensaje de error y devuelve a la EMPRESA al menú anterior.

CASO DE USO: MODIFICAR CUPONES.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: C2.

Objetivo en Contexto:

Permite a un administrador cambiar la cantidad de cupones que tiene un particular con una determinada empresa.

Actor principal: Administrador.

Actores secundarios: Base de Datos de Usuarios.

Qué datos usa: Datos sobre los cupones acumulados por el particular.

Precondiciones:

El Administrador debe haber iniciado sesión.

Postcondiciones:

Éxito: Modifica los cupones acumulados por el particular seleccionado para la empresa seleccionada. Muestra un mensaje de éxito.

Fallo: Se muestra un mensaje de error, vuelve al menú anterior.

Flujo principal

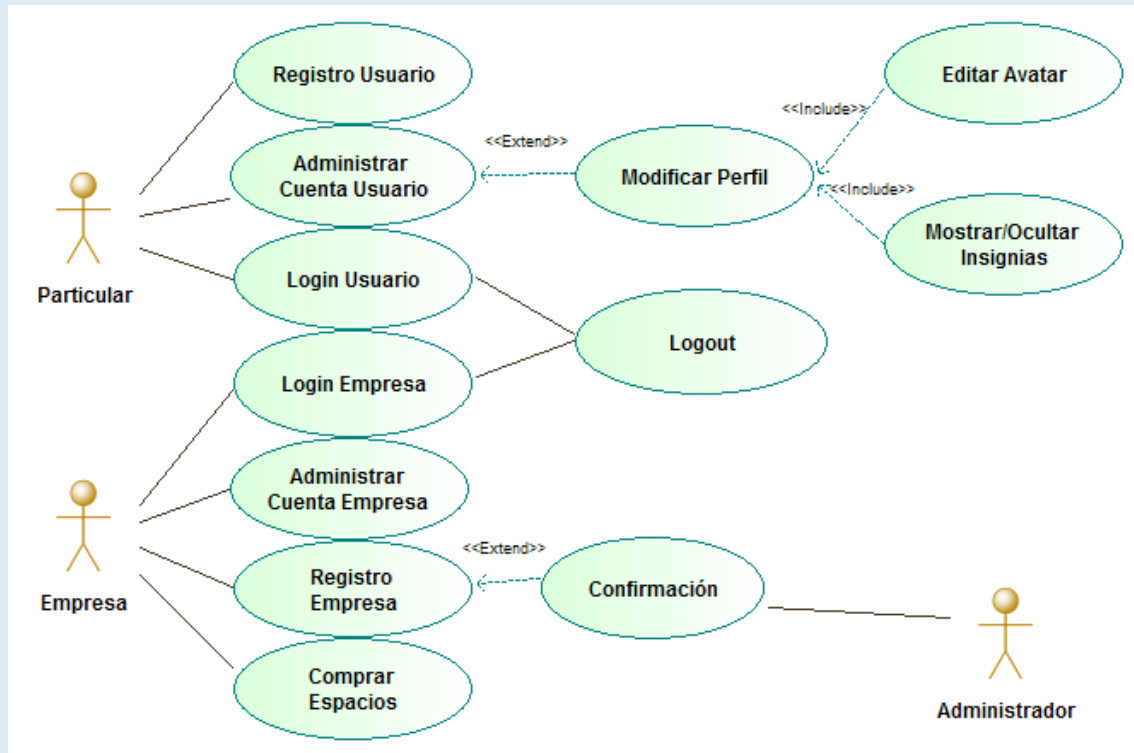
1. El ADMINISTRADOR selecciona al particular y la opción “modificar cupones”.
2. El ADMINISTRADOR selecciona la empresa para la cual modificará el número de cupones.
3. El ADMINISTRADOR introduce la nueva cantidad de cupones y selecciona “confirmar”.
4. El SISTEMA solicita a la base de datos que modifique la cantidad de cupones correspondiente.
5. El SISTEMA comprueba que la cantidad ha sido actualizada con éxito.
6. El SISTEMA muestra un mensaje de éxito.

Flujos secundarios

- 5-a. Si la cantidad no ha sido actualizada correctamente, el SISTEMA muestra un mensaje de error y devuelve al ADMINISTRADOR al menú anterior.

GESTIÓN

DIAGRAMA DE CASOS DE USO



CASOS DE USO

CASO DE USO: ADMINISTRAR CUENTA (EMPRESAS).

Nombre del AUTOR: Oscar Díaz Ribagorda

Identificador: GE2.

Objetivo en Contexto:

La empresa puede consultar y cambiar los datos de su cuenta, como, su correo (identificación), su contraseña, su nombre, su página web, etc.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas.

Qué datos usa: Datos registrados de la empresa que quiere administrar la cuenta.

Precondiciones:

La empresa tiene que estar registrada.

La empresa tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: Los datos de la empresa se cambian y se muestran los datos actualizados.

Fallo: Se indica que los datos no han podido ser cambiados y se sale de la administración de cuenta.

Flujo principal

1. La EMPRESA selecciona la opción del menú que pone “Mi Cuenta”.
2. El SISTEMA muestra los datos guardados de la cuenta de esa empresa.
3. La EMPRESA pulsa el botón de “Editar Cuenta”.
4. La EMPRESA sobrescribe los campos que quiera cambiar.
5. La EMPRESA selecciona la opción de “Guardar Cambios”.
6. El SISTEMA muestra de nuevo los datos de la cuenta, con los nuevos cambios.

Flujos secundarios

- 3-a. En el caso que la EMPRESA no quiera modificar los campos, pulsa el botón “Atrás” y el SISTEMA muestra de nuevo el menú de inicio.
- 4-a. Si la EMPRESA quiere cambiar el campo “Contraseña”, esta debe introducir la contraseña actual para proceder a cambiar el campo.
- 4-b. Si la EMPRESA decide no cambiar ningún campo, pulsa el botón de “Atrás”, y se cancela la actualización de los datos.
- 6-a. En el caso de que no se hayan podido cambiar los campos, se cancela la actualización y se muestra un mensaje de error.

CASO DE USO: COMPRAR ESPACIOS

Nombre del AUTOR: Lucas de Torre Barrio

Identificador: GE3.

Objetivo en Contexto:

Comprar espacios para poder publicar más ofertas.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas.

Qué datos usa: Ofertas de la empresa y espacios disponibles de la empresa.

Precondiciones:

La empresa tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: La empresa consigue espacios, teniendo la opción de publicar más ofertas o sorteos.

Fallo: No se compran los espacios y se muestra un mensaje de error.

Flujo principal

1. La EMPRESA selecciona la opción del menú que pone “Comprar espacios” en el menú.
2. El SISTEMA muestra una pantalla con los espacios disponibles de la EMPRESA y pide seleccionar cuántos espacios se quieren comprar (y actualizando el precio).
3. La EMPRESA selecciona la opción de “Comprar n espacios”.
4. El SISTEMA muestra la pasarela de pago y la empresa introduce los datos de pago.
5. La empresa adquiere los espacios.

Flujos secundarios

- 3-a. La EMPRESA decide no comprar espacios y pulsa “atrás”.
- 5-b. El SISTEMA detecta que los datos de pago son incorrectos y muestra un mensaje de error.
- 6-b. Se vuelven a pedir los datos de pago.

CASO DE USO: CONFIRMACIÓN DE REGISTROS.

Nombre del autor: Pablo Hidalgo Palencia

Identificador: GC

Objetivo en Contexto: Dado un nuevo registro, de una empresa, se comprueba que se trata de una empresa real, para que no haya lugar a fraudes ni ofertas falsas. En caso de particulares no es necesario puesto que estos no pueden hacer ofertas.

Actor principal: Administrador.

Actores secundarios: Base de Datos.

Qué datos usa: Información de las empresas ya registradas.

Precondiciones: Todavía no se ha confirmado si la empresa es real.

Postcondiciones:

Éxito: Queda confirmada la permanencia de la cuenta de la empresa en la aplicación.

Fallo: Se elimina la cuenta de esa empresa, y consecuentemente sale de la base de datos.

Flujo principal:

1. El administrador busca los datos de la empresa en cuestión en la base de datos.
2. El administrador recaba información de la empresa a la que se hace referencia en esos datos, ya sea por Internet, con llamadas, datos externos...
3. El administrador comprueba que los datos se corresponden con los que se habían aportado y que, en efecto, se trata de una empresa real que pretende publicitarse mediante ofertas en la aplicación, y que acepta los términos de uso.
4. Se marca la empresa en la base de datos para que quede claro que ya se ha realizado el proceso de confirmación de esta empresa.

Flujos secundarios/alternativos:

3-a. Si se consiguen datos suficientes como para saber que no se trata de una empresa real o que ha habido algún error y la empresa no pretende continuar con esa cuenta, se procede a la eliminación de la cuenta y su retirada de la base de datos.

CASO DE USO: REGISTRO EMPRESA.

Nombre del autor: Pablo Hidalgo Palencia

Identificador: GE0

Objetivo en Contexto: registro de una nueva empresa en el sistema, porque la empresa lo solicita

Actor principal: Empresa.

Actores secundarios: Base de Datos.

Qué datos usa: Información de las empresas ya registradas.

Precondiciones: La empresa quiere registrarse

Postcondiciones:

Éxito: Se registra la nueva empresa con sus datos y se modifica convenientemente la aplicación para que esta nueva empresa esté dentro.

Fallo: Se muestran mensajes aclaratorios de los errores y se permite reintentar el registro.

Flujo principal:

1. La empresa hace clic en "Registrarse como empresa", dentro del menú de registros
2. La empresa introduce su correo electrónico y contraseña y confirma que acepta los términos y condiciones del registro.
3. El sistema comprueba que estos datos son correctos.
4. El sistema busca en la base de datos de empresas si ya estaba registrada esta empresa.
5. En caso de que no estuviera, se procede a introducir a esa empresa en la base de datos.
6. Se permite entrar a la empresa, ya registrada, en la aplicación

Flujos secundarios/alternativos:

- 3-a. Si algún campo no es válido (correo electrónico mal escrito o no existente, por ejemplo), se le muestra el error y se vuelven a pedir esos datos.
- 3-b. Si no se aceptan los términos de uso, no se puede registrar, y se le muestra la advertencia.
- 4-a. En caso de que la empresa ya estuviera registrada, se notifica y se vuelve a los pasos anteriores para pedir nuevos datos.

CASO DE USO: REGISTRO USUARIO

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: GP0

Objetivo en Contexto:

El usuario hace click en el botón Registrarse.

El usuario introduce con teclado (físico o virtual) su correo electrónico y su contraseña para la aplicación.

El usuario confirma su intención de registrarse.

Actor principal: Usuario.

Actores secundarios: Base de Datos.

Qué datos usa: Información de usuarios registrados.

Precondiciones:

El usuario solicita registrarse.

El usuario introduce sus datos de acceso.

Postcondiciones:

Éxito: Se registra el nuevo usuario y el usuario entra en la aplicación.

Fallo: Mensajes de Error y vuelta a pedir datos.

Flujo principal

1. El USUARIO hace click en el botón “Registrarse”.
2. El USUARIO introduce su correo electrónico y su contraseña en los campos para ello.
3. El USUARIO presiona el botón “confirmar” o la tecla Enter.
4. El SISTEMA comprueba la validez de los datos introducidos.
5. El SISTEMA pregunta a la BASE DE DATOS si existe un usuario con ese correo.
6. La BASE DE DATOS responde al SISTEMA.
7. El SISTEMA solicita a la BASE DE DATOS que registre al nuevo usuario, si no está registrado ya ese correo.
8. La BASE DE DATOS registra al nuevo usuario con su contraseña.
9. El SISTEMA da acceso al usuario a la aplicación.

Flujos secundarios

5-a.

Alguno de los dos campos (correo o contraseña) es inválido (está vacío o no cumple los requisitos de seguridad, el correo no es de la forma [a@b.c](#), etc.).

6-a.

El SISTEMA muestra un Mensaje de ERROR, permite volver a introducirlos.

7-b.

El correo YA está registrado en la BASE DE DATOS.

8-b.

El SISTEMA muestra un Mensaje de ERROR, permite volver a introducir los datos.

9-c.

Se produce un error en el registro de los datos de usuario en la BASE DE DATOS.

10-c.

El SISTEMA muestra un Mensaje de ERROR, permite volver a introducir los datos.

CASO DE USO: LOGIN USUARIO

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: GP1

Objetivo en Contexto:

El usuario introduce con teclado (físico o virtual) su correo electrónico y su contraseña para la aplicación.

El usuario hace clic en el botón Entrar o presiona la tecla Enter.

Actor principal: Usuario.

Actores secundarios: Base de Datos.

Qué datos usa: Información de usuarios registrados.

Precondiciones:

El usuario introduce sus datos de acceso.

El usuario solicita identificarse para acceder a la aplicación.

Postcondiciones:

Éxito: El usuario entra en la aplicación.

Fallo: Mensajes de Error y vuelta a pedir datos.

Flujo principal

1. El USUARIO introduce su correo electrónico y su contraseña en los campos para ello.
2. El USUARIO presiona la tecla Enter o el botón de “Entrar”.
3. El SISTEMA solicita a la BASE DE DATOS la contraseña del correo electrónico introducido.
4. La BASE DE DATOS devuelve la contraseña al SISTEMA, si el correo está registrado.
5. El SISTEMA coteja la contraseña introducida con la devuelta por la BASE DE DATOS.
6. El SISTEMA da acceso al usuario a la aplicación si las dos contraseñas coinciden

Flujos secundarios

3-a.

Alguno de los dos campos (correo o contraseña) es vacío.

4-a.

El SISTEMA muestra un Mensaje de ERROR, permite volver a introducirlos.

4-b.

El correo no está registrado en la BASE DE DATOS.

5-b.

El SISTEMA muestra un Mensaje de ERROR, permite volver a introducir los datos.

6-c. La contraseña proporcionada por la BASE DE DATOS y la introducida por el USUARIO no coinciden.

7-c. El SISTEMA muestra un Mensaje de ERROR, permite volver a introducir los datos.

CASO DE USO: EDITAR AVATAR

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: GPPA

Objetivo en Contexto:

El usuario selecciona la opción de editar su avatar.

El usuario modifica su avatar con las opciones básicas y los complementos que haya desbloqueado.

El usuario confirma las modificaciones.

Actor principal: Usuario

Actores secundarios: Base de Datos de Usuarios.

Qué datos usa: Información de usuarios registrados y sus cupones.

Precondiciones:

El usuario debe haber iniciado sesión.

Postcondiciones:

Éxito: Se modifica el avatar del usuario en la aplicación.

Fallo: Se muestra un mensaje de error y se mantiene el avatar anterior.

Flujo principal

1. El USUARIO hace click en el botón “Editar Avatar”.
2. El SISTEMA solicita a la base de datos, la información sobre los cupones del usuario.
3. El SISTEMA muestra al usuario el menú de edición de avatares.
4. El USUARIO edita su avatar sin poder utilizar los complementos (gorras, elementos decorativos, etc.) que requieran una cantidad de cupones que no tenga con la empresa correspondiente.
5. El USUARIO confirma que ha terminado de editar su perfil con el botón “Confirmar”.
6. El SISTEMA envía la información a la base de datos de usuarios para que actualice el avatar.
7. El SISTEMA muestra un mensaje de éxito y devuelve al usuario al menú de modificar perfil.

Flujos secundarios

3-a.

Si la base de datos no devuelve estos datos, el SISTEMA mostrará un mensaje de error.

5-b.

El USUARIO selecciona la opción “Cancelar”.

6-b.

El SISTEMA devuelve al usuario al menú de modificar perfil.

CASO DE USO: MOSTRAR/OCULTAR INSIGNIAS

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: GPPI

Objetivo en Contexto:

El usuario selecciona la opción de editar sus insignias visibles.

El usuario selecciona para cada insignia a la que tenga derecho si quiere o no mostrarla en su perfil.

El usuario confirma las modificaciones.

Actor principal: Usuario

Actores secundarios: Base de Datos de Usuarios.

Qué datos usa: Información de usuarios registrados y sus insignias.

Precondiciones:

El usuario debe haber iniciado sesión. El usuario debe tener alguna insignia otorgada.

Postcondiciones:

Éxito: Se modifican las insignias visibles en el perfil del usuario.

Fallo: Se muestra un mensaje de error y se mantiene la configuración.

Flujo principal

1. El USUARIO hace click en el botón “Mostrar/Ocultar Insignias”.
2. El SISTEMA solicita a la base de datos, la información sobre las insignias del usuario.
3. El SISTEMA muestra al usuario una lista con sus insignias.
4. El USUARIO selecciona para cada una si quiere mostrarla u ocultarla.
5. El USUARIO confirma que ha terminado de editar su perfil con el botón “Confirmar”.
6. El SISTEMA envía la información a la base de datos de usuarios para que actualice el perfil.
7. El SISTEMA muestra un mensaje de éxito y devuelve al usuario al menú de modificar perfil.

Flujos secundarios

3-a.

Si la base de datos no devuelve estos datos, el SISTEMA mostrará un mensaje de error.

5-b.

El USUARIO selecciona la opción “Cancelar”.

6-b.

El SISTEMA devuelve al usuario al menú de modificar perfil.

CASO DE USO: LOGOUT

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: GL

Objetivo en Contexto:

El usuario o empresa solicita cerrar su sesión (haciendo click en el botón “Desconectarse”).

Actor principal: Usuario/Empresa

Actores secundarios: SISTEMA

Qué datos usa: N/A

Precondiciones:

El usuario o empresa deben tener la sesión abierta.

El usuario o empresa solicita cerrar su sesión.

Postcondiciones:

Éxito: El usuario o empresa salen de la aplicación.

Fallo: N/A.

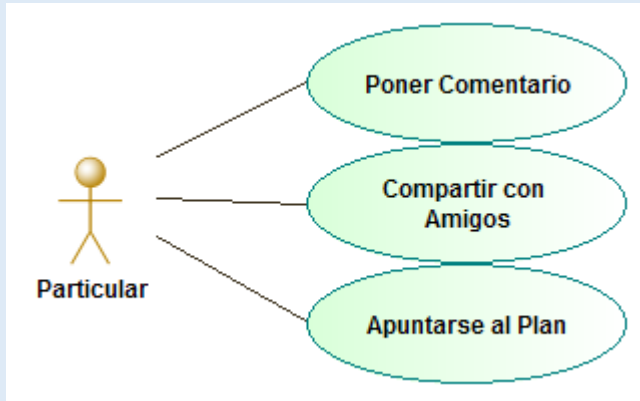
Flujo principal

1. El USUARIO/La EMPRESA solicita cerrar su sesión.
2. El SISTEMA revoca el acceso a la aplicación al USUARIO/EMPRESA.

Flujos secundarios

SOCIAL

DIAGRAMA DE CASOS DE USO



CASOS DE USO

CASO DE USO: PONER COMENTARIO.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: SPC.

Objetivo en Contexto:

Publicar un comentario en el perfil de una Empresa.

Actor principal: Particular

Actores secundarios: Base de Datos de Empresas y Usuarios.

Qué datos usa: Información de la empresa y datos del usuario.

Precondiciones:

El particular debe haber iniciado sesión en la aplicación.

Postcondiciones:

Éxito: Se publica el comentario en el perfil de la empresa bajo el apartado “Comentarios”.

Fallo: Se muestra un mensaje de error en la publicación del comentario.

Flujo principal

1. El PARTICULAR selecciona la opción “Publicar Comentario” dentro del perfil de la empresa.
2. El SISTEMA muestra al particular el menú de escritura de comentarios.
3. El PARTICULAR escribe su comentario y selecciona la opción de “Publicar Comentario”.
4. El SISTEMA solicita a la base de datos el número total de cupones del usuario.
5. El SISTEMA calcula la relevancia del comentario en base a los cupones totales del particular y lo publica en el perfil de la empresa mandando la información a la base de datos.
6. El SISTEMA notifica el éxito de la operación al particular.

Flujos secundarios

- 5-a. En el caso de que el SISTEMA no reciba la información necesaria, mostrará un mensaje de error al Particular.
- 6-b. Si el SISTEMA no recibe confirmación de la base de datos de que el comentario se ha publicado con éxito, mostrará un mensaje de error al particular.

CASO DE USO: COMPARTIR CON AMIGOS.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: SCA.

Objetivo en Contexto:

Generar un mensaje con un enlace a una oferta que pueda enviarse por correo electrónico y enviar el correo.

Actor principal: Particular

Actores secundarios: Base de Datos de Ofertas y Usuarios.

Qué datos usa: Información de la Oferta.

Precondiciones:

El particular debe haber iniciado sesión en la aplicación. El Particular debe estar apuntado a la oferta.

Postcondiciones:

Éxito: Se envía un correo electrónico a la dirección introducida por el Particular.

Fallo: Se muestra un mensaje de error a la hora de compartir.

Flujo principal

1. El PARTICULAR selecciona la opción “Compartir con Amigos” para una oferta que ha aceptado.
2. El SISTEMA muestra al particular el menú de elección de destinatario.
3. El PARTICULAR escribe una dirección de correo electrónico y selecciona la opción de “Compartir Oferta”.
4. El SISTEMA pregunta al Particular si quiere enviar la oferta a otra persona.
5. El PARTICULAR selecciona la opción de “No”.
6. El SISTEMA solicita a la base de datos la descripción de la oferta y el nombre del particular.
7. El SISTEMA genera un correo electrónico con la descripción de la oferta y un enlace a la misma, invitando a participar en nombre del particular.
8. El SISTEMA envía el correo a todas las direcciones introducidas por el usuario.

Flujos secundarios

- 5-a. El PARTICULAR selecciona la opción de “Sí”.
- 6-a. Volvemos a 2.
- 7-b. Si el SISTEMA no recibe los datos de la oferta o del usuario de la base de datos, mostrará un mensaje de error al particular.
- 8-c. Si se produce un error a la hora de generar el correo electrónico, el SISTEMA muestra un mensaje de error al particular.

CASO DE USO: APUNTARSE AL PLAN.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: SAP.

Objetivo en Contexto:

Publicar un mensaje en la página de una oferta indicando que el particular va a participar en ella.

Actor principal: Particular

Actores secundarios: Base de datos de Ofertas y de Usuarios

Qué datos usa: Información del particular.

Precondiciones:

El particular debe haber iniciado sesión en la aplicación. El Particular debe haber aceptado la oferta

Postcondiciones:

Éxito: Se publica el mensaje del particular indicando además cuando y donde quiere montar su plan.

Fallo: Presenta un mensaje de error al particular y no publica nada.

Flujo principal

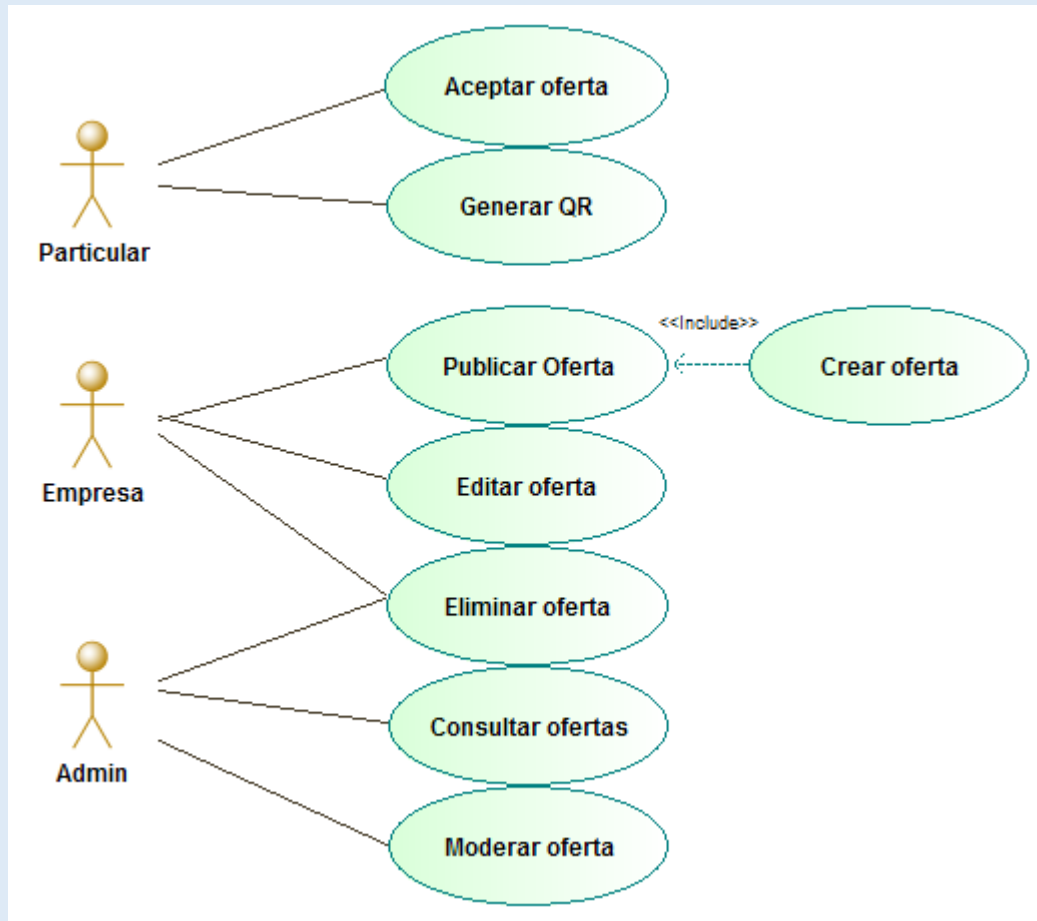
1. El PARTICULAR selecciona la opción “Apuntarse al Plan” dentro de la página de la oferta.
2. El SISTEMA consulta a la base de datos si el particular ha aceptado la oferta.
3. El SISTEMA muestra el menú de apuntarse al plan. Consta de un desplegable para el día y la hora, una casilla para indicar el lugar y una caja de texto para escribir más detalles.
4. El PARTICULAR escribe los detalles de su plan, los datos de donde y cuando se llevará a cabo y selecciona la opción de “Publicar Plan”.
5. El SISTEMA solicita a la base de datos el nombre y avatar del particular.
6. El SISTEMA publica el plan del particular en la página de la oferta, mandando la información a la base de datos.
7. El SISTEMA notifica el éxito de la operación al particular.

Flujos secundarios

- 3-a. Si el particular no ha aceptado la oferta correspondiente, el SISTEMA mostrará un mensaje de error indicándolo y le devolverá a la página de la oferta.
- 5-b. Si el particular no ha indicado el lugar o la hora de su plan, el SISTEMA mostrará un mensaje de error pidiéndole que lo haga y le devolverá al menú de apuntarse al plan.
- 5-c. Si el particular selecciona la opción “Cancelar”, el SISTEMA le devolverá a la página de la oferta.
- 6-d. Si el SISTEMA no recibe los datos solicitados de forma correcta, mostrará un mensaje de error y devolverá al particular al menú de apuntarse al plan.
- 7-2. Si el SISTEMA no recibe un mensaje de la base de datos confirmado el éxito de la publicación, mostrará un mensaje de error y devolverá al particular al menú de apuntarse al plan.

OFERTAS

DIAGRAMA DE CASOS DE USO



CASOS DE USO

CASO DE USO: CREAR OFERTA.

Nombre del AUTOR: Oscar Díaz Ribagorda

Identificador: CO.

Objetivo en Contexto:

Creación de una nueva oferta por parte de la empresa con todos los campos necesarios: Nombre, descripción de la oferta, condiciones, lugar, foto, cupón(opcional).

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas.

Qué datos usa: Datos sobre la oferta que introduce la empresa, datos sobre la empresa.

Precondiciones:

La empresa tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: La oferta es creada y se guarda en una lista de ofertas de la empresa (no se publica en la lista de ofertas general todavía).

Fallo: Se muestra un mensaje de error en la creación de oferta y se sale de la opción de crear oferta.

Flujo principal

1. La EMPRESA selecciona la opción del menú que pone “Crear Oferta”.
2. El SISTEMA muestra una plantilla con los datos a rellenar por la EMPRESA incluyendo los opcionales.
3. La EMPRESA introduce los campos obligatorios de la oferta, y los campos opcionales que desee.
4. La EMPRESA selecciona la opción de “Crear Oferta”.
5. El SISTEMA comprueba que todos los campos introducidos sean correctos.
6. El SISTEMA muestra la oferta creada, dentro de la lista de ofertas de la empresa.

Flujos secundarios

- 2-a. En el caso que la EMPRESA no quiera crear la oferta, selecciona el botón de “Cancelar”, el SISTEMA cancela la creación de la oferta y muestra de nuevo el menú principal.
- 3-a. Si el SISTEMA reconoce que un campo introducido es erróneo, muestra un mensaje de “Campo incorrecto” y pide de nuevo la información del campo.
- 5-a. Si los datos introducidos no son los suficientes, o no son correctos, no se crea la oferta. Se muestra un mensaje en el campo que falta o incorrecto y se queda en la pantalla de introducir los campos.

CASO DE USO: EDITAR OFERTA.

Nombre del AUTOR: Oscar Díaz Ribagorda

Identificador: EO.

Objetivo en Contexto:

Dada una oferta (no publicada) ya creada de una empresa, se pueden editar los campos de la oferta siempre y cuando los nuevos campos sigan cumpliendo los requisitos.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas.

Qué datos usa: Datos de la oferta seleccionada (dentro de las ofertas de la empresa).

Precondiciones:

La empresa tiene que tener la sesión iniciada.

Tiene que tener al menos una oferta (no publicada) creada.

Postcondiciones:

Éxito: La oferta es modificada con éxito y muestra la oferta con los cambios realizados.

Fallo: Se muestra un mensaje de error, y no se modifica la oferta.

Flujo principal

1. La EMPRESA selecciona la opción del menú que pone "Mis Ofertas".
2. La EMPRESA selecciona una de las ofertas (sin publicar).
3. El SISTEMA muestra la oferta seleccionada por pantalla.
4. La EMPRESA selecciona el botón de "Editar Oferta".
5. La EMPRESA modifica los campos que desee.
6. El SISTEMA comprueba que todos los campos sean correctos.
7. El SISTEMA muestra la oferta modificada, dentro de la lista de ofertas de la empresa.

Flujos secundarios

- 5-a. En el caso que la EMPRESA no quiera editar la oferta, selecciona el botón de "Cancelar", el SISTEMA cancela la edición de la oferta y muestra la lista de ofertas de la empresa.
- 5-b. Si el SISTEMA reconoce que un campo introducido es erróneo, muestra un mensaje de "Campo incorrecto" y pide de nuevo la información del campo.
- 6-a. Si los datos introducidos no son los suficientes, o no son correctos, no se modifica la oferta. Se muestra un mensaje en el campo que falta o incorrecto y se queda en la pantalla de introducir los campos.

CASO DE USO: ELIMINAR OFERTA (EMPRESA).

Nombre del AUTOR: Lucas de Torre Barrio

Identificador: EOE.

Objetivo en Contexto:

Elimina una oferta creada.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Ofertas.

Qué datos usa: Ofertas de la empresa.

Precondiciones:

La empresa tiene que tener la sesión iniciada y una oferta creada.

Postcondiciones:

Éxito: La oferta es eliminada.

Fallo: Se muestra un mensaje de error.

Flujo principal

1. La EMPRESA selecciona una de las ofertas de la lista.
2. El SISTEMA muestra una pantalla con la oferta y la EMPRESA selecciona "Eliminar oferta".
3. El SISTEMA elimina la oferta.

Flujos secundarios

- 3-a. El SISTEMA muestra un mensaje de error si no pudo dar de baja la oferta.

CASO DE USO: PUBLICAR OFERTA.

Nombre del AUTOR: Lucas de Torre Barrio

Identificador: PO.

Objetivo en Contexto:

Publicar una oferta creada previamente siempre y cuando la empresa tenga espacios disponibles.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas.

Qué datos usa: Ofertas de la empresa y espacios disponibles de la empresa.

Precondiciones:

La empresa tiene que tener la sesión iniciada, una oferta creada y, al menos, un espacio libre para publicar la oferta.

Postcondiciones:

Éxito: La oferta es publicada y se resta uno a los espacios disponibles de la empresa para publicar ofertas.

Fallo: Se muestra un mensaje de error en la publicación de la oferta, originado por la falta de espacios para ofertas de la empresa.

Flujo principal

1. La EMPRESA selecciona la opción del menú que pone "Publicar oferta" en una oferta sin publicar.
2. El SISTEMA muestra una pantalla con los espacios disponibles de la EMPRESA preguntando si quiere publicar la oferta.
3. La EMPRESA selecciona la opción de "Publicar Oferta".
4. El SISTEMA comprueba que la empresa tuviera espacios disponibles.
5. El SISTEMA publica la oferta y resta uno a los espacios de la empresa.

Flujos secundarios

- 5-a. En el caso que la EMPRESA no tenga espacios disponibles, el SISTEMA no publica la oferta.
- 6-a. El SISTEMA muestra un mensaje de error indicando que no hay espacios disponibles.

CASO DE USO: CONSULTAR OFERTAS.

Nombre del autor: Pablo Hidalgo Palencia

Identificador: E10

Objetivo en Contexto: que el administrador se pueda informar de las ofertas que hay en la aplicación.

Actor principal: Administrador.

Actores secundarios: Base de Datos.

Qué datos usa: Información de las ofertas publicadas por cada empresa.

Precondiciones: el administrador desea informarse sobre ofertas.

Postcondiciones:

Éxito: Consigue acceder a la información que solicita y modificarla en caso necesario.

Fallo:

Flujo principal:

1. El administrador hace clic en "Consultar ofertas", dentro de su menú.
2. El sistema muestra todas las ofertas que tiene la aplicación, ordenadas o filtradas dependiendo de distintos parámetros que el administrador puede elegir, y permite que pinche en cualquiera de ellas para informarse.

Flujos secundarios/alternativos:

2-a. En caso de que el administrador pinche sobre alguna oferta, accederá a sus datos: empresa que lo ha publicado, descripción, usuarios relacionados con ella (porque les ha gustado o la han visto, por ejemplo...). En caso de que lo considere necesario, por el contenido o por otros motivos, podrá eliminarla, cambiarla... o mandar un mensaje a la empresa que la ha publicado para informar.

CASO DE USO: ELIMINAR OFERTA (ADMIN).

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: EOA.

Objetivo en Contexto:

Elimina una oferta creada por una empresa.

Actor principal: Administrador.

Actores secundarios: Base de Datos de Ofertas.

Qué datos usa: Ofertas publicadas.

Precondiciones:

El administrador debe tener iniciada sesión.

Postcondiciones:

Éxito: La oferta es eliminada y la empresa notificada de por qué.

Fallo: Se muestra un mensaje de error.

Flujo principal

1. El ADMINISTRADOR selecciona una de las ofertas de la lista.
2. El SISTEMA muestra una pantalla con la oferta y el ADMINISTRADOR selecciona "Eliminar oferta".
3. El SISTEMA muestra al administrador un cuadro de texto para que explique por qué se ha eliminado la oferta.
4. El ADMINISTRADOR aporta la información de por qué ha eliminado la oferta.
5. El SISTEMA elimina la oferta y envía un mensaje al correo electrónico de la empresa informando de por qué se ha dado de baja su oferta.

Flujos secundarios

- 5-a. El SISTEMA muestra un mensaje de error si no pudo dar de baja la oferta.
- 5-b. El SISTEMA muestra un mensaje de error si no pudo enviar el correo.

CASO DE USO: ACEPTAR OFERTA

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: AO.

Objetivo en Contexto:

Acepta una oferta creada por una empresa.

Actor principal: Particular.

Actores secundarios: Base de Datos de Ofertas, Base de Datos de Usuarios.

Qué datos usa: Ofertas publicadas.

Precondiciones:

El particular debe tener iniciada sesión.

Postcondiciones:

Éxito: El particular acepta la oferta, dándole derecho a generar un QR.

Aumenta el número de particulares que han aceptado la oferta.

Fallo: Se muestra un mensaje de error.

Flujo principal

1. El PARTICULAR selecciona una de las ofertas de la lista.
2. El SISTEMA muestra una pantalla con la oferta y el PARTICULAR selecciona "Aceptar oferta".
3. El SISTEMA aumenta en 1 el contador de "Han Aceptado" en la página de la oferta y guarda el dato en la base de datos de ofertas.
4. El SISTEMA marca la oferta como aceptada por el particular en la base de datos de USUARIOS.
5. El SISTEMA ofrece al particular la opción de generar un código QR para poder validar su oferta.

Flujos secundarios

- 3-a. El SISTEMA muestra un mensaje de error si no pudo actualizar el dato.
- 4-b. El SISTEMA muestra un mensaje de error si no pudo marcar como aceptada.

CASO DE USO: GENERAR QR (OFERTA)

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: GQRO.

Objetivo en Contexto:

Genera un código QR para una oferta aceptada por el particular.

Actor principal: Particular.

Actores secundarios: Base de Datos de Usuarios.

Qué datos usa: Ofertas aceptadas por el usuario.

Precondiciones:

El particular debe tener iniciada sesión. El particular debe haber aceptado la oferta.

Postcondiciones:

Éxito: Se genera un código QR que permite canjear la oferta.

Fallo: Se muestra un mensaje de error.

Flujo principal

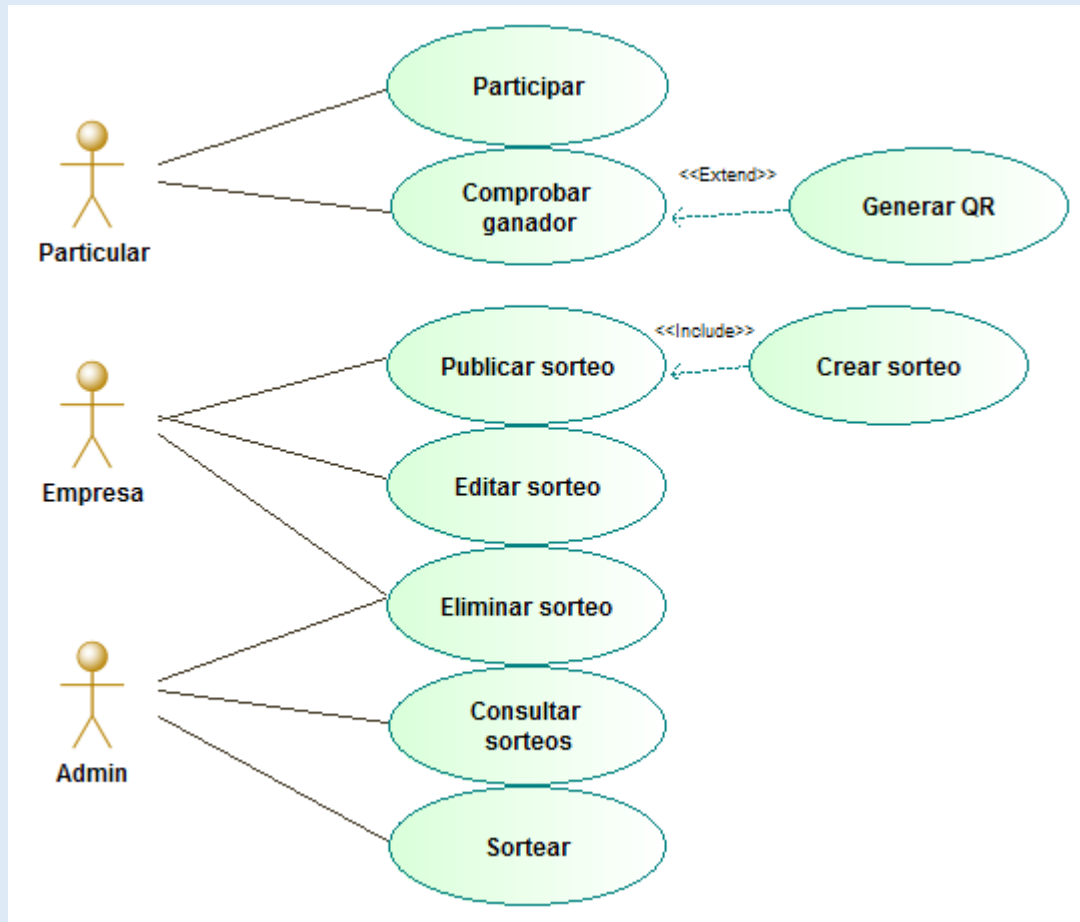
1. El PARTICULAR selecciona una oferta que haya aceptado.
2. El SISTEMA comprueba con la base de datos que efectivamente la oferta ha sido aceptada por el particular.
3. El SISTEMA genera un código QR válido para la oferta, lo muestra por pantalla y pide al particular que haga una captura de pantalla.

Flujos secundarios

- 3-a. El SISTEMA muestra un mensaje de error si no pudo comprobarlo.
- 4-b. El SISTEMA muestra un mensaje de error si no pudo generar el código.

SORTEOS

DIAGRAMA DE CASOS DE USO



CASOS DE USO

CASO DE USO: CREAR SORTEO.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: CS.

Objetivo en Contexto:

Creación de un nuevo sorteo por parte de la empresa con todos los campos necesarios: Nombre, descripción del sorteo, criterios de acceso, lugar, foto, etc.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas y de Sorteos.

Qué datos usa: Datos sobre el sorteo que introduce la empresa, datos sobre la empresa.

Precondiciones:

La empresa tiene que tener la sesión iniciada.

Postcondiciones:

Éxito: El sorteo es creado y se guarda en una lista de sorteos de la empresa (no se publica en la lista de sorteos general todavía).

Fallo: Se muestra un mensaje de error en la creación de sorteo y se sale de la opción de crear sorteo.

Flujo principal

7. La EMPRESA selecciona la opción del menú que pone "Crear Sorteo".
8. El SISTEMA muestra una plantilla con los datos a rellenar por la EMPRESA incluyendo los opcionales.
9. La EMPRESA introduce los campos obligatorios del sorteo, y los campos opcionales que desee.
10. La EMPRESA selecciona la opción de "Crear Sorteo".
11. El SISTEMA comprueba que todos los campos introducidos sean correctos.
12. El SISTEMA muestra el sorteo creado, dentro de la lista de sorteos de la empresa.

Flujos secundarios

- 2-a. En el caso que la EMPRESA no quiera crear el sorteo, selecciona el botón de "Cancelar", el SISTEMA cancela la creación del sorteo y muestra de nuevo el menú principal.
- 3-a. Si el SISTEMA reconoce que un campo introducido es erróneo, muestra un mensaje de "Campo incorrecto" y pide de nuevo la información del campo.
- 5-a. Si los datos introducidos no son los suficientes, o no son correctos, no se crea el sorteo. Se muestra un mensaje en el campo que falta o incorrecto y se queda en la pantalla de introducir los campos.

CASO DE USO: EDITAR SORTEO.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: ES.

Objetivo en Contexto:

Dado un sorteo (no publicado) ya creado de la empresa, se pueden editar los campos y requisitos de acceso al sorteo siempre y cuando los nuevos cambios sigan cumpliendo los requisitos.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas y de sorteos.

Qué datos usa: Datos del sorteo seleccionado (dentro de los sorteos de la empresa).

Precondiciones:

La empresa tiene que tener la sesión iniciada.

Tiene que tener al menos un sorteo (no publicado) creado.

Postcondiciones:

Éxito: El sorteo es modificado con éxito y muestra el sorteo con los cambios realizados.

Fallo: Se muestra un mensaje de error, y no se modifica el sorteo.

Flujo principal

8. La EMPRESA selecciona la opción del menú que pone "Mis Sorteos".
9. La EMPRESA selecciona un de los sorteos (sin publicar).
10. El SISTEMA muestra el sorteo seleccionado por pantalla.
11. La EMPRESA selecciona el botón de "Editar Sorteo".
12. La EMPRESA modifica los campos que desee.
13. El SISTEMA comprueba que todos los campos sean correctos.
14. El SISTEMA muestra el sorteo modificado, dentro de la lista de sorteos de la empresa.

Flujos secundarios

- 5-a. En el caso que la EMPRESA no quiera editar el sorteo, selecciona el botón de "Cancelar", el SISTEMA cancela la edición del sorteo y muestra la lista de sorteos de la empresa.
- 5-b. Si el SISTEMA reconoce que un campo introducido es erróneo, muestra un mensaje de "Campo incorrecto" y pide de nuevo la información del campo.
- 6-a. Si los datos introducidos no son los suficientes, o no son correctos, no se modifica el sorteo. Se muestra un mensaje en el campo que falta o incorrecto y se queda en la pantalla de introducir los campos.

CASO DE USO: ELIMINAR SORTEO (EMPRESA).

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: ESE.

Objetivo en Contexto:

Elimina un sorteo creado por la empresa.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Sorteos.

Qué datos usa: Sorteos de la empresa.

Precondiciones:

La empresa tiene que tener la sesión iniciada y un sorteo creado.

Postcondiciones:

Éxito: El sorteo es eliminado.

Fallo: Se muestra un mensaje de error.

Flujo principal

4. La EMPRESA selecciona uno de los sorteos de la lista.
5. El SISTEMA muestra una pantalla con el sorteo y la EMPRESA selecciona “Eliminar sorteo”.
6. El SISTEMA elimina el sorteo.

Flujos secundarios

- 3-a. El SISTEMA muestra un mensaje de error si no pudo dar de baja el sorteo.

CASO DE USO: PUBLICAR SORTEO.

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: PS.

Objetivo en Contexto: Publicar un sorteo creado previamente, siempre y cuando la empresa tenga espacios disponibles.

Actor principal: Empresa.

Actores secundarios: Base de Datos de Empresas y de sorteos.

Qué datos usa: Sorteos de la empresa y espacios disponibles de la empresa.

Precondiciones:

La empresa tiene que tener la sesión iniciada, un sorteo creado y, al menos, un espacio libre para publicar la oferta.

Postcondiciones:

Éxito: El sorteo es publicado y se resta uno a los espacios disponibles de la empresa para publicar sorteos.

Fallo: Se muestra un mensaje de error en la publicación del sorteo.

Flujo principal

6. La EMPRESA selecciona la opción del menú que pone “Publicar sorteo” en un sorteo sin publicar.
7. El SISTEMA muestra una pantalla con los espacios disponibles de la EMPRESA preguntando si quiere publicar el sorteo.
8. La EMPRESA selecciona la opción de “Publicar sorteo”.
9. El SISTEMA comprueba con la base de datos que la empresa tuviera espacios disponibles.
10. El SISTEMA publica el sorteo y resta uno a los espacios de la empresa.

Flujos secundarios

- 5-a. En el caso que no se pueda acceder a la base de datos de sorteos o empresas, el SISTEMA mostrará un mensaje de error.
- 5-b. En el caso que la EMPRESA no tenga espacios disponibles, el SISTEMA no publica el sorteo.
- 6-b. El SISTEMA muestra un mensaje de error indicando que no hay espacios disponibles.

CASO DE USO: SORTEAR

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: S0.

Objetivo en Contexto: Sortea manualmente un sorteo antes de su fecha de vencimiento.

Actor principal: Administrador.

Actores secundarios: Base de Datos de Sorteos.

Qué datos usa: Participantes en el sorteo.

Precondiciones:

El sorteo debe estar publicado. El Administrador debe haber iniciado sesión.

Postcondiciones:

Éxito: Extrae un ganador entre los participantes y le envía un correo electrónico informando de ello.

Fallo: Se muestra un mensaje de error.

Flujo principal

1. La ADMINISTRADOR selecciona la opción "Sortear ahora" en un sorteo sin concluir.
2. El SISTEMA accede a la base de datos para obtener una lista de participantes en el sorteo.
3. El SISTEMA extrae un ganador de entre los participantes.
4. El SISTEMA envía un correo electrónico al ganador.

Flujos secundarios

- 3-a. En el caso que no se pueda acceder a la base de datos, el SISTEMA muestra un mensaje de error.
- 3-b. En el caso de que no haya nadie apuntado, el SISTEMA muestra un mensaje de error indicando que no hay ningún participante.
- 4-c. Si no se pudo mandar el correo, el SISTEMA lo comunica con un mensaje de error.

CASO DE USO: COMPROBAR GANADOR

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: CG.

Objetivo en Contexto: Permitir que el Particular consulte quien ha ganado un sorteo en el que participaba.

Actor principal: Particular

Actores secundarios: Base de Datos de Sorteos y de Usuarios.

Qué datos usa: Participantes en el Sorteo, identificador del particular y perfil del ganador.

Precondiciones:

El particular debe haber participado en el sorteo. El sorteo debe haber concluido.

Postcondiciones:

Éxito: Muestra al particular quién ha ganado ese sorteo y lo elimina de la lista de sorteos en los que participa. Permite acceder al perfil del ganador o generar un QR (solo en caso de que haya ganado el particular).

Fallo: Se muestra un mensaje de error.

Flujo principal

1. El PARTICULAR selecciona "Comprobar Ganador" para un determinado sorteo.
2. El SISTEMA solicita a la base de datos quien ha ganado el sorteo y la lista de sorteos donde participa el particular.
3. El SISTEMA muestra por pantalla un enlace al perfil del ganador del sorteo.

Flujos secundarios

- 2-a. En caso de que el SISTEMA no pueda acceder a la base de datos mostrará un mensaje de error.
- 2-b. En caso de que el particular no participe en el sorteo, el SISTEMA mostrará un mensaje en el que pinga "Solo puedes consultar ganadores de sorteos en los que hayas participado".
- 2-c. En caso de que el sorteo no haya concluido, el SISTEMA mostrará un mensaje en el que pinga "Solo puedes consultar ganadores de sorteos que hayan finalizado".
- 3-d. En caso de que el particular sea el ganador del sorteo, el SISTEMA mostrará un cuadro con un mensaje de felicitación y la opción de generar un QR para canjear el premio.

CASO DE USO: CONSULTAR SORTEOS.

Nombre del autor: Daniel Loscos Barroso

Identificador: CS

Objetivo en Contexto: que el administrador se pueda informar de los sorteos que hay en la aplicación.

Actor principal: Administrador.

Actores secundarios: Base de Datos de Sorteos.

Qué datos usa: Información de los sorteos publicados por cada empresa.

Precondiciones: El administrador debe haber iniciado sesión.

Postcondiciones:

Éxito: Consigue acceder a la información que solicita y modificarla en caso necesario.

Fallo: Muestra un mensaje de error.

Flujo principal:

3. El ADMINISTRADOR hace clic en "Consultar sorteos", dentro de su menú.
4. El SISTEMA muestra todos los sorteos que tiene la aplicación, ordenados o filtrados dependiendo de distintos parámetros que el administrador puede elegir, y permite que pinche en cualquiera de ellos para informarse.

Flujos secundarios/alternativos:

2-a. Si el SISTEMA no puede mostrar los sorteos, aparecerá un mensaje de error.

3-b. En caso de que el ADMINISTRADOR pinche sobre algún sorteo, accederá a sus datos: empresa que lo ha publicado, descripción, usuarios que participan en él, etc. En caso de que lo considere necesario, por el contenido o por otros motivos, podrá eliminarlo, cambiarlo o mandar un mensaje a la empresa que lo ha publicado para informar.

CASO DE USO: ELIMINAR SORTEO (ADMIN).

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: ESA.

Objetivo en Contexto: Elimina un sorteo creado por una empresa.

Actor principal: Administrador.

Actores secundarios: Base de Datos de Ofertas.

Qué datos usa: Ofertas publicadas.

Precondiciones:

El administrador debe tener iniciada sesión.

Postcondiciones:

Éxito: El sorteo es eliminado y la empresa notificada de por qué.

Fallo: Se muestra un mensaje de error.

Flujo principal

6. El ADMINISTRADOR selecciona uno de los sorteos de la lista.
7. El SISTEMA muestra una pantalla con el sorteo y el ADMINISTRADOR selecciona "Eliminar sorteo".
8. El SISTEMA muestra al administrador un cuadro de texto para que explique por qué se ha eliminado el sorteo.
9. El ADMINISTRADOR aporta la información de por qué ha eliminado el sorteo.
10. El SISTEMA elimina el sorteo y envía un mensaje al correo electrónico de la empresa informando de por qué se ha dado de baja su sorteo.

Flujos secundarios

- 5-a. El SISTEMA muestra un mensaje de error si no pudo dar de baja el sorteo.
- 5-b. El SISTEMA muestra un mensaje de error si no pudo enviar el correo.

CASO DE USO: PARTICIPAR

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: AS.

Objetivo en Contexto:

Apunta al Particular en un determinado sorteo si cumple los requisitos.

Actor principal: Particular.

Actores secundarios: Base de Datos de Sorteos, Base de Datos de Usuarios.

Qué datos usa: Características del sorteo, participantes en el sorteo, características y cupones del usuario.

Precondiciones:

El particular debe tener iniciada sesión. El particular debe cumplir los requisitos marcados para entrar en el sorteo.

Postcondiciones:

Éxito: El particular entra en el sorteo. Aumenta el número de participantes en el sorteo.

Fallo: Se muestra un mensaje de error.

Flujo principal

6. El PARTICULAR selecciona "Participar" en la página del sorteo.
7. El SISTEMA solicita a la base de datos los requisitos del participante y las características del usuario.
8. El SISTEMA coteja la información de la base de datos.
9. El SISTEMA añade al particular a la lista de participantes y el sorteo a la lista de sorteos del particular.

Flujos secundarios

- 3-a. El SISTEMA muestra un mensaje de error si no pudo obtener los datos.
- 4-b. El SISTEMA muestra un mensaje de error si el participante no cumple los requisitos del sorteo.

CASO DE USO: GENERAR QR (SORTEO)

Nombre del AUTOR: Daniel Loscos Barroso

Identificador: GQRS.

Objetivo en Contexto:

Genera un código QR para el ganador de un sorteo y se lo muestra.

Actor principal: Particular.

Actores secundarios: Base de Datos de Usuarios y de Sorteos.

Qué datos usa: Identificador del usuario y ganador del sorteo.

Precondiciones:

El particular debe tener iniciada sesión. El particular debe haber ganado el sorteo.

Postcondiciones:

Éxito: Se genera un código QR que permite canjear el premio.

Fallo: Se muestra un mensaje de error.

Flujo principal

4. El PARTICULAR selecciona un sorteo en el que haya participado.
5. El SISTEMA comprueba con la base de datos que efectivamente el sorteo ha sido ganado por el particular.
6. El SISTEMA genera un código QR válido para la recompensa, lo muestra por pantalla y pide al particular que haga una captura de pantalla.

Flujos secundarios

- 3-a. El SISTEMA muestra un mensaje de error si no pudo comprobarlo.
- 3-b. Si el PARTICULAR no ha ganado el sorteo, el SISTEMA se lo notifica.
- 4-c. El SISTEMA muestra un mensaje de error si no pudo generar el código.

REQUISITOS FUNCIONALES (CRC'S)

En este apartado están todas las CRC's ordenadas por subsistemas (La imagen tiene vínculos a los diferentes subsistemas para poder acceder a ellos):



OFERTAS

Nombre de Clase: *ListaOfertas*SuperClases: *Engine*SubClases: *Oferta*

Atributos	Clases colaboradoras
Lista (Array de objetos <i>Oferta</i>): Guarda una lista completa de ofertas	<i>Oferta</i>
TamLista (entero): Nos indica cuántos elementos tiene esta lista	
Buscador (Buscador): Permitirá efectuar búsquedas con filtros	<i>Buscador</i>
Métodos	Interacción con otras clases
añadir: aÑade un elemento a la lista	
eliminar: quita un elemento ya existente	
buscar: efectúa una búsqueda con ciertos parámetros	<i>Engine, Buscador</i>
ordenar: ordena ofertas conforme a los gustos de un usuario o ciertos parámetros más	<i>Engine, Particular, Buscador</i>

Nombre de la clase: *Oferta*

SuperClases: *ListaOfertas*

SubClases: --

Atributos	Clases colaboradoras
Título (Cadena): Nombre de la oferta, con limitación en su extensión	
Descripción (Cadena/Texto): Nos informa de todo lo que un usuario puede querer saber de dicha oferta	
FechaVencimiento (fecha, se pueden usar tipos predefinidos en librerías): Día a partir del cual la oferta no estará vigente	
Imagen (tipo compatible de imagen): Resume gráficamente la oferta	
Oferente (Empresa): Empresa que la publicó	<i>Empresa</i>
Métodos	Interacción con otras clases
Oferta: constructor	
modificarOferta: permite modificar la descripción	<i>Engine</i>
getOferente: nos permitirá saber quién la publicó	<i>Empresa</i>
getFecha: devuelve la fecha de vencimiento de la oferta	
cambiarFecha: permite modificar la fecha de vencimiento	<i>Engine</i>
cambiarImagen: nos permite poner otra imagen	

SORTEOS

Nombre de Clase : ListaSorteosSuperClases: **Engine**SubClases: **Sorteo, Buscador**

Atributos	Clases colaboradoras
Lista (Array de objetos Sorteo): Guarda una serie de sorteos	
TamLista (entero): Tamaño de la lista	
Buscador (objeto de tipo Buscador): Permitirá efectuar búsquedas con filtros	<i>Buscador</i>
Métodos	Interacción con otras clases
añadir: mete otro sorteo a la lista	
eliminar: quita un sorteo de la lista	
buscar: Busca un sorteo en la lista conforme a un perfil indicado	<i>Buscador</i>
ordenar: Permite ordenar la lista conforme a varios parámetros o los gustos del usuario	<i>Engine, Particular, Buscador</i>

Nombre de Clase : *Sorteo*

SuperClases: *ListaSorteos*

SubClases: --

Atributos	Clases colaboradoras
Título (Cadena): Nombre del sorteo	
Descripción (Cadena/Texto): Da toda la información a los usuarios sobre un determinado sorteo	
Fecha (fecha, a buscar tipo en librerías): Día en que se celebrará el sorteo	
NumParticipantes (entero): Número de personas apuntadas al sorteo en el momento	
Participantes (ListaParticulares) Los que ya están apuntados al sorteo	<i>ListaParticulares</i>
Imagen (algún formato de imagen compatible): Ayudará en la descripción y publicitación del sorteo	
Empresa (empresa): Empresa que publicó el sorteo	<i>Empresa</i>
Métodos	Interacción con otras clases
Sorteo: constructor	
modificar: permite modificar la descripción	<i>Engine</i>
getEmpresa: nos permitirá saber quién la publicó	
getFecha: devuelve la fecha de celebración del sorteo	
cambiarFecha: permite modificar la fecha del sorteo	<i>Engine</i>
cambiarImagen: nos permite poner otra imagen	

BUSCADOR

Nombre de Clase: <i>Buscador</i>	
SuperClases: <i>ListaSorteos, ListaOfertas, ListaUsuarios</i>	
SubClases: --	
Función general: <i>Efectuará búsquedas de diferente tipo (todavía por determinar) sobre listas</i>	
Atributos	Clases colaboradoras
Lista: <i>Se buscará sobre esa lista</i>	
Métodos	Interacción con otras clases

GESTIÓN

Nombre de Clase: *Engine*

SuperClases: --

SubClases: *ListaUsuarios, ListaOfertas, ListaSorteos*

Función general:

Será el motor principal de la aplicación, de donde provienen todas sus clases, manejando por tanto todo el entramado de funcionalidades. Lidera la interacción con el usuario, pero delegará en sus subclases a la hora de efectuar todas estas tareas. Lidera también la interacción entre diferentes clases de la aplicación.

Atributos	Clases colaboradoras
ListaEmpresas (<i>ListaEmpresas</i>): Lista de empresas que tienen cuenta en la aplicación	<i>ListaOfertas</i>
ListaParticulares (<i>ListaParticulares</i>): Lista de todos los particulares de la aplicación	<i>ListaParticulares</i>
ListaOfertas (<i>ListaOfertas</i>): Lista de todas las ofertas publicadas en el momento	<i>ListaOfertas</i>
ListaSorteos (<i>ListaSorteos</i>): Guarda todos los sorteos que existen en la aplicación en el momento	<i>ListaSorteos</i>
Usuario (<i>particular/empresa</i>): Nos indica quién es el usuario que está utilizando la aplicación	<i>Empresa, Particular</i>
Métodos	Interacción con otras clases
mostrarOfertas : Ordenados por las preferencias del usuario	<i>ListaOfertas</i>
mostrarSorteos : Ordenados por las preferencias del usuario	<i>ListaSorteos</i>
login : permite iniciar sesión	<i>ListaParticulares, Particular, ListaEmpresas, Empresa</i>
logout : termina la sesión	

registrarse: <i>Crea nuevo usuario</i>	<i>ListaParticulares, Particular, ListaEmpresas, Empresa</i>
mostrarPerfil: <i>muestra la información del usuario</i>	
ver: <i>Tanto ofertas como sorteos, para ver su información</i>	<i>ListaOfertas, ListaSorteos, Oferta, Sorteo</i>
publicarOferta: <i>añade a la lista de ofertas publicadas</i>	<i>ListaOfertas</i>
publicarSorteo: <i>añade a la lista de sorteos publicados</i>	<i>ListaSorteos</i>
comprarEspacios: <i>permite comprar espacio para publicar nuevas ofertas y sorteos</i>	<i>Por determinar, necesaria pasarela de pago</i>
getTexto: <i>pide un cierto texto necesario para completar datos en la creación o modificación de ciertos objetos</i>	

Nombre de Clase: *ListaUsuarios*SuperClases: ***Engine***SubClases: ***ListaParticulares, ListaEmpresas***Función general: *Se trata de una clase abstracta de la que derivan los dos tipos de usuarios***Atributos****Clases colaboradoras**Las clases *ListaEmpresas* y *ListaParticulares* derivan de esta clase**Métodos****Interacción con otras clases*****añadir******eliminar******buscar***

Nombre de Clase: *ListaParticulares*

SuperClases: *ListaUsuarios*

SubClases: *Particular*

Atributos	Clases Colaboradoras
Lista (Array de objetos <i>Particular</i>): <i>Lista de particulares de la aplicación</i>	<i>Particular</i>
TamLista (entero): <i>Número de particulares en la lista</i>	
Buscador (Objeto de tipo buscador): <i>Nos permitirá buscar en la lista conforme a ciertos filtros</i>	<i>Buscador</i>
Métodos	Interacción con otras clases
añadir: <i>Añade un Particular a la lista existente</i>	
eliminar: <i>Quita a un Particular de la lista</i>	
buscar: <i>Busca a un particular conforme a los datos que se le indiquen</i>	<i>Buscador</i>

Nombre de Clase: Particular	
SuperClases: ListaParticulares	
SubClases: Plan, DatosParticular	
Atributos	Clases colaboradoras
Datos (DatosParticular): Datos del usuario	<i>DatosParticular</i>
ListaInsignias (Array de objetos Insignia): lista que guarda las insignias (premios de la aplicación a los usuarios por utilizarla)	<i>Insignia</i>
ListaGustadas (ListaOfertas): lista de ofertas que le han gustado	<i>ListaOfertas</i>
Métodos	Interacción con otras clases
añadirOferta: Añadir oferta a la lista de gustadas	<i>ListaOfertas</i>
eliminarOferta: Eliminar oferta de la lista de gustadas	<i>ListaOfertas</i>
cambiarNombre: Cambia el nombre de usuario	<i>Engine, DatosParticular</i>
cambiarContraseña: Cambia la contraseña del usuario	<i>Engine, DatosParticular</i>
getDatos: Devolver la información del particular	<i>DatosParticular</i>
añadirInsignia: Añade una insignia a la lista (en principio no hará falta eliminarlas)	
Particular: Constructor para crear Particular a partir de datos que se aporten	

Nombre de la clase: *DatosParticular*SuperClases: **Particular**

SubClases: --

Atributos**Clases colaboradoras****Nombre** (Cadena):
Nombre del usuario**Contraseña** (Cadena):
Contraseña del usuario**Métodos****Interacción con otras clases****setNombre:**
cambia el nombre**setContraseña:**
cambia la contraseña**getNombre:**
devuelve el nombre**getContraseña:**
devuelve la contraseña

Nombre de la clase: *ListaEmpresas*

SuperClases: *ListaUsuarios*

SubClases: *Empresa*

Atributos	Clases colaboradoras
Lista (Array de objetos Empresa): Lista de Empresas de la aplicación	<i>Empresa</i>
TamLista (entero): Número de empresas en la lista	
Buscador (Objeto de tipo buscador): Nos permitirá buscar en la lista conforme a ciertos filtros	<i>Buscador</i>
Métodos	Interacción con otras clases
añadir: añade una empresa a la lista	
eliminar: quita una empresa de la lista	
busca : busca empresas con un perfil parecido al que le indicamos	<i>Buscador</i>

Nombre de la clase: *Empresa*

SuperClases: *ListaEmpresas*

SubClases: --

Atributos	Clases colaboradoras
Datos (objeto de tipo <i>DatosEmpresa</i>): Encapsula todos los datos que tenemos de una empresa	<i>DatosEmpresa</i>
numOfertas (entero): Número de ofertas publicadas	
ListaOfertasPublicadas (objeto <i>ListaOfertas</i>): Lista de ofertas que una empresa tiene actualmente publicadas	<i>ListaOfertas</i>
Tokens (entero): Número de espacios disponibles	
numSorteos (entero): número de sorteos actualmente publicados	
ListaSorteos (objeto <i>ListaSorteos</i>): Sorteos que la empresa tiene publicados en el momento	<i>ListaSorteos</i>
Métodos	Interacción con otras clases
Empresa: Constructor a partir de datos iniciales (<i>DatosEmpresa</i>)	<i>DatosEmpresa</i>
crearOferta: pide los datos necesarios para que se pueda generar una oferta para su posterior publicación	<i>Engine, Oferta</i>
modificarOferta: cambia una oferta existente	<i>Engine, Oferta</i>
publicarOferta: pone una oferta existente a disposición de los usuarios de la aplicación (esta acción gasta tokens)	<i>Engine, ListaOfertas</i>
comprarEspacios: se paga por ellos a través de la aplicación	Por determinar, se necesitaría alguna pasarela de pago
crearSorteo: da la opción de generar un nuevo sorteo	<i>Engine, Sorteo</i>
publicarSorteo: permite poner a disposición	<i>Engine, ListaSorteos</i>

<i>de los usuarios un sorteo</i>	
modificarSorteo: <i>cambia un sorteo ya existente</i>	<i>Engine, Sorteo</i>
eliminarSorteo: <i>Hace que un sorteo deje de existir</i>	<i>Engine, ListaSorteos</i>
eliminarOferta: <i>Quita una oferta de la aplicación</i>	<i>Engine, ListaOfertas</i>

Nombre de la clase: DatosEmpresa	
SuperClases: Particular	
SubClases: --	
Atributos	Clases colaboradoras
Nombre (Cadena): Nombre del usuario	
Contraseña (Cadena): Contraseña del usuario	
Ocupación (Cadena): Indica a lo que se dedica la empresa (que debe ser verificado por los administradores)	
Métodos	Interacción con otras clases
setNombre : cambia el nombre	
setContraseña : cambia la contraseña	
getNombre : devuelve el nombre	
getContraseña : devuelve la contraseña	

Nombre de la clase: *Insignia*

SuperClases: **Particular**

SubClases: --

Atributos	Clases colaboradoras
Nombre (Cadena): Cada insignia tendrá asociada un nombre	
Imagen (formato compatible imagen): La apariencia de la insignia	
Empresa (Empresa): Indica de qué empresa proviene la insignia, en caso de que así sea	<i>Empresa</i>
Motivo (Cadena): Nos informa de cuál es el motivo por el cual se dará esta insignia a un usuario	
Métodos	Interacción con otras clases
mostrarMotivo : devuelve el motivo para ganar la insignia en un formato para ser mostrado	
mostrarInsignia : devuelve la insignia para poder mostrarla	

SOCIAL

Nombre de Clase: Comentarios**SuperClases:** *Oferta, Sorteo.***SubClases:** *Comentario.*

Atributos	Clases colaboradoras
Comentarios (Lista de objetos Comentario): Guarda todos los comentarios respecto a un mismo tema	<i>Comentario</i>
TamLista (entero): Número de comentarios existentes	
Métodos	Interacción con otras clases
añadir: mete un nuevo comentario en la lista	
eliminar: quita de la lista un comentario	

Nombre de Clase: **Comentario**

SuperClases: **Comentarios**

SubClases: --

Atributos	Clases colaboradoras
Usuario (Empresa/Particular): Guarda quién publicó ese comentario	<i>ListaEmpresas, ListaUsuarios</i>
Contenido (Cadena/Texto) : Guarda el mensaje del comentario	
RespuestaA (Comentario): Nos informa de a qué otro mensaje está respondiendo (en caso de que así sea)	
Métodos	Interacción con otras clases
getUsuario: nos devuelve quién publicó ese comentario	
cambiarContenido: Nos permite modificar el texto del comentario	<i>Engine</i>
responder: da la opción de crear un nuevo comentario en respuesta a uno dado	<i>Engine, Comentarios</i>
Comentario: Constructor de objetos comentario en base a los atributos de los comentarios	<i>Engine</i>

Nombre de Clase: <i>Plan</i>	
SuperClases: <i>Particular</i>	
SubClases: --	
Atributos	Clases colaboradoras
Oferta (Oferta): Oferta en base a la cual se hace un plan	<i>Ofertas</i>
TipoPlan (Tipo enumerado que puede tomar los valores Público y Privado): Nos indica si el plan se hace solo con amigos o con todos los usuarios de la aplicación que quieran	
NumParticipantes (entero): Número de personas que se han apuntado al plan ya	
ListaParticipantes (ListaParticulares): En ella están todos los que se han apuntado	<i>ListaParticulares</i>
Métodos	Interacción con otras clases
añadirParticipante: mete un participante más a la lista	<i>ListaParticulares</i>
eliminarParticipante: quita a un participante ya apuntado	<i>ListaParticulares</i>
getParticipantes: Devuelve el número de participantes del plan	

CUPONES

Nombre de Clase: <i>Cupón</i>	
SuperClases: <i>Oferta</i>	
SubClases: --	
Atributos	Clases colaboradoras
Nombre (Cadena): Nombre de la oferta de la que proviene	
QR (código QR): Código que se necesitará para validar el cupón	
Métodos	Interacción con otras clases
getNombre: devuelve el nombre de la oferta de la que viene	
getQR: devuelve el código del cupón	
Cupón: Constructor que necesita nombre y QR	

REQUISITOS DE RENDIMIENTO

Sabemos que esto es una mera estimación, pero creemos que nuestros servidores deberían ser capaces de almacenar 100.000.000 cuentas y como mínimo soportar 1.000.000 usuarios simultáneos.

REQUISITOS LÓGICOS DE LA BASE DE DATOS

Toda la información, tanto a la referente a las cuentas como a los sorteos y ofertas será almacenada en una base de datos que podrá manejar solicitudes de distintos usuarios de forma simultánea

RESTRICCIONES DE DISEÑO

El diseño de este proyecto está sujeto a los estándares de programación y diseño de diagramas (UML estándar). También hay que tener en cuenta que la aplicación debería ocupar y consumir lo menos posible sin perder de vista un manejo cómodo y agradable, al estar diseñada para el teléfono móvil.

ATRIBUTOS DEL SISTEMA SOFTWARE

Entre los atributos fundamentales de nuestra aplicación encontramos:

<u>Fiabilidad:</u> La aplicación debe someterse a rigurosas pruebas que aseguren su correcto funcionamiento en todos los casos.	<u>Disponibilidad:</u> La aplicación estará disponible de forma gratuita para el mayor público posible por lo que estará disponible en Android e iOS.
<u>Seguridad:</u> Toda la información respectiva a las cuentas tanto de usuarios particulares como la de las empresas estará protegida y no será distribuida a terceros.	<u>Fácil Mantenimiento:</u> La compañía garantizará el mantenimiento y correcto uso de la aplicación mediante el uso de los usuarios administradores.

APÉNDICES