

10 DE MAYO DE 2020



APLICACIÓN VALENCIA EXPLORA

DOCUMENTO PROYECTO DEFINITIVO

JUAN ALMEIDA – JAVIER GONZÁLEZ – ALEJANDRO TINOCO

1ºDAW
C.I.F.P Mislata

ÍNDICE

Contenido

DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE (BD).....2

DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE GLOBAL4

DIAGRAMA DE CASOS DE USO6

DIAGRAMA DE CLASES DE LA APLICACIÓN.....8

DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE (BD)

Valencia Explora (1ºDAW-1ºGIAT)

Hoy, a día 6 de marzo de 2020, el Cliente Valencia Explora S.L, cuya representación en la reunión por parte de ésta la compusieron **Dragana Onofrei, Rebeca Serrano y Daniel Ruiz** Acordaron con el grupo desarrollador cuyos integrantes, todos presentes en la reunión, lo componen **Alejandro Tinoco, Juan Almeida y Francisco Javier González**.

Lo siguiente:

La aplicación tiene un principal objetivo: Gestionar la compra de packs turísticos, conformados por actividades que los usuarios objetivo elegirán libremente, dentro del conjunto de actividades que ofrezca la propia aplicación. La ciudad sobre la que versarán los packs será Valencia.

Los packs serán creados por los usuarios, los cuales tienen asignados un rol, pudiendo ser administrador o cliente. A parte de poder explorar la aplicación, el cliente podrá generar, modificar y eliminar packs, además de gestionar sus datos de usuario. El administrador por otra parte podrá crear, modificar y eliminar tanto perfiles de clientes o como actividades.

Tanto administradores como clientes se les asignará un código identificativo único, consecutivo al usuario anterior.

En cuanto a los clientes, estos deben acceder con su e-mail, único, y contraseña. Si no están registrados, deben aportar obligatoriamente su nombre, e-mail, contraseña y número de contacto. Otros datos opcionales durante el registro son su tarjeta de crédito, apellido, y gustos personales. Aunque posea opcionalidad, la tarjeta de crédito será obligatorio implementarla si el usuario se dispone a realizar el pago de un pack.

Los clientes pueden crear los packs que quieran, aunque cada pack será exclusivo del cliente que lo crea. Por ello, a cada pack creado también se le asignará un código identificativo único y consecutivo. Adicionalmente, cada pack registrado deberá tener un nombre, una posible descripción, y las fechas de inicio y fin del propio pack. Todo ello según las directrices del propio usuario. La aplicación automáticamente registrará la fecha de creación del pack.

Cada pack contiene actividades, componiéndose cada pack de una o más actividades. Asimismo, una actividad puede pertenecer a muchos packs.

Las actividades se identifican por un código identificativo único, además de un nombre, una URL y una posible descripción e Imagen. Para cada actividad incluida en el pack, se deberá de guardar también el precio, la cantidad de personas asociadas a la reserva y la fecha y hora de inicio y final de la actividad, así como la duración.

Se establecen 4 TIPOS de actividades: “localizaciones”, “restauración”, “alojamientos” y “Tours”. Estos tipos quedan registrados por un código único y un nombre. Sin embargo, el cliente expresa la necesidad de ofrecerle la capacidad de poder añadir o eliminar tipos.

Asimismo, cada actividad se asocia a un subtipo. La función del subtipo será la de especializar más la actividad. Cada subtipo se identificará un código identificativo único, un nombre y un pequeño logotipo. A modo de ejemplo con este último apartado, una actividad dentro del grupo “localización” puede tener un subtipo “para niños” o “aire libre”, mientras que otra perteneciente a “restauración” puede tener un subtipo asociado como “comida mediterránea” “asiática”, entre otros.

Para concluir, una actividad tendrá siempre asignado un subtipo, aunque un subtipo puede pertenecer a varias actividades. En cuanto a los tipos de actividades, cada uno de ellos puede contener varios subtipos, pero un subtipo solo puede pertenecer a un tipo de actividad.

Sin nada más que añadir, las partes acuerdan estas características iniciales, sujetas a posibles modificaciones que podrían ser remunerables.

DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE GLOBAL

Valencia Explora (1ºDAW-1ºGIAT)

Hoy, a día 6 de marzo de 2020, el Cliente Valencia Explora S.L, cuya representación en la reunión por parte de ésta la compusieron **Dragana Onofrei, Rebeca Serrano y Daniel Ruiz** Acordaron con el grupo desarrollador cuyos integrantes, todos presentes en la reunión, lo componen **Alejandro Tinoco, Juan Almeida y Francisco Javier González**.

Lo siguiente:

Valencia Explora es una empresa turística que desea gestionar “Packs” donde se incluyen diferentes actividades. La empresa está orientada un público adulto de poder adquisitivo medio-alto, que requiere una carta de servicios amplia, diversa y a la vez de calidad. Los clientes muy probablemente van acompañados de niños, por lo que sus actividades también deben de ir dedicadas a este nicho de clientes.

El objetivo principal de la aplicación solicitada es la **gestión simple, intuitiva y eficiente por parte del cliente** (visitante o registrado) **de “Packs Turísticos”, compuestos de una serie de tipos de actividades o “recursos”** (“Localizaciones”, “Restaurantes”, “Alojamientos”, “Tours”). Estos recursos pueden combinarse, y no es necesario contratar todos los recursos en cada Pack. Cada tipo de recurso a su vez se compone de subtipos que lo caracterizan. Estos subtipos especifican más cada actividad (por ejemplo, en una localización especificaría si ésta es de entorno religioso, aire libre, festivo...).

Los usuarios, que pueden ser **clientes o administradores**, pueden explorar la aplicación libremente (en su entorno de cliente), pero únicamente los registrados son los habilitados para guardar como favoritos estos packs, añadir tarjetas, gustos de cliente o números de contacto, dando opción a añadir algún otro apartado.

La aplicación constará de una serie de “escenas”, las cuales, de manera muy concreta son:

1. **Un menú de inicio**, en la que se especifique la entrada a la aplicación por parte de un usuario visitante o uno registrado. Si se elige la opción de registrado, la aplicación se redirige a una ventana de log-in.
2. **Un menú principal** en la que el usuario pueda interaccionar con los distintos tipos de recursos. El menú principal debería mostrar un carrusel con imágenes asociadas a los gustos establecidos por el usuario previamente. En caso de que no se incluyan, el carrusel debería mostrar imágenes aleatorias de los productos ofertados. Por otra parte, se deberían de mostrar preferentemente productos en oferta.
3. **Recursos**: Donde se muestran en forma de “cuadrados” los diferentes productos (localizaciones, alojamientos, tours o restaurantes), en la que se incluyen una imagen y su nombre. Se acordó posteriormente incluir logotipos de las características principales de cada localización. Cada recurso irá por separado; es decir, van en escenas independientes donde no se deben de juntar. En el tipo “Tours” se debe de incluir específicamente su valoración como logotipo, sin incidir en detalles sobre éste.
4. **Recursos detallados**: Tras clicar los recursos correspondientes, se especifica que ese recurso se “expanda” y muestre sus detalles (horarios, descripción, coste, menú, url...). Será en esta nueva ventana detallada donde se de la opción de contratar el producto,

seleccionar características especiales (hora de reserva, personas...) y añadirlo al carrito. En el apartado "Tours" se debe detallar la valoración con comentarios de los usuarios de la propia aplicación.

5. **Log-in:** donde el usuario registrado podrá introducir su nombre de usuario/email y su contraseña. En el caso de que el usuario sea invitado tiene la opción de dirigirse a una escena de registro. El log-in se especifica que se haga también con un log-in automático mediante cuenta de Facebook o Google.
6. **Registro:** Típico formulario de registro con datos clave y opcionales.
7. **Área Cliente:** Ventana donde el cliente registrado puede modificar su perfil, revisar con detalle sus packs preferidos-comprados o bien añadir gustos que repercutirán en la experiencia del usuario posteriormente.
8. **Carrito:** Ventana donde se especifican los productos seleccionados anteriormente, dando la opción al pago.
9. **Admin:** Ventana única para el Propietario-Administrador. En esta ventana, en la que se debe hacer log-in con un usuario con plenos privilegios, se debe facilitar la creación, modificación, eliminación de packs para usuarios registrados, así como características de sus perfiles. Por otra parte, debe permitir la adición de nuevos tipos y subtipos de actividades.

OTROS:

- Se indica la opcionalidad de añadir una pequeña ventana que indique el tiempo en Valencia el día en el que se usa la aplicación y los días siguientes, incluyendo iconos. Se debe de incluir información de valor para padres (*family friendly*).
- El usuario se debe mostrar en todo momento durante la navegación en la esquina superior-derecha.
- La aplicación debe mostrar una pequeña ventana de ayuda, donde se especifique número y mail de la empresa para que ésta, en primera instancia, ayude al cliente.

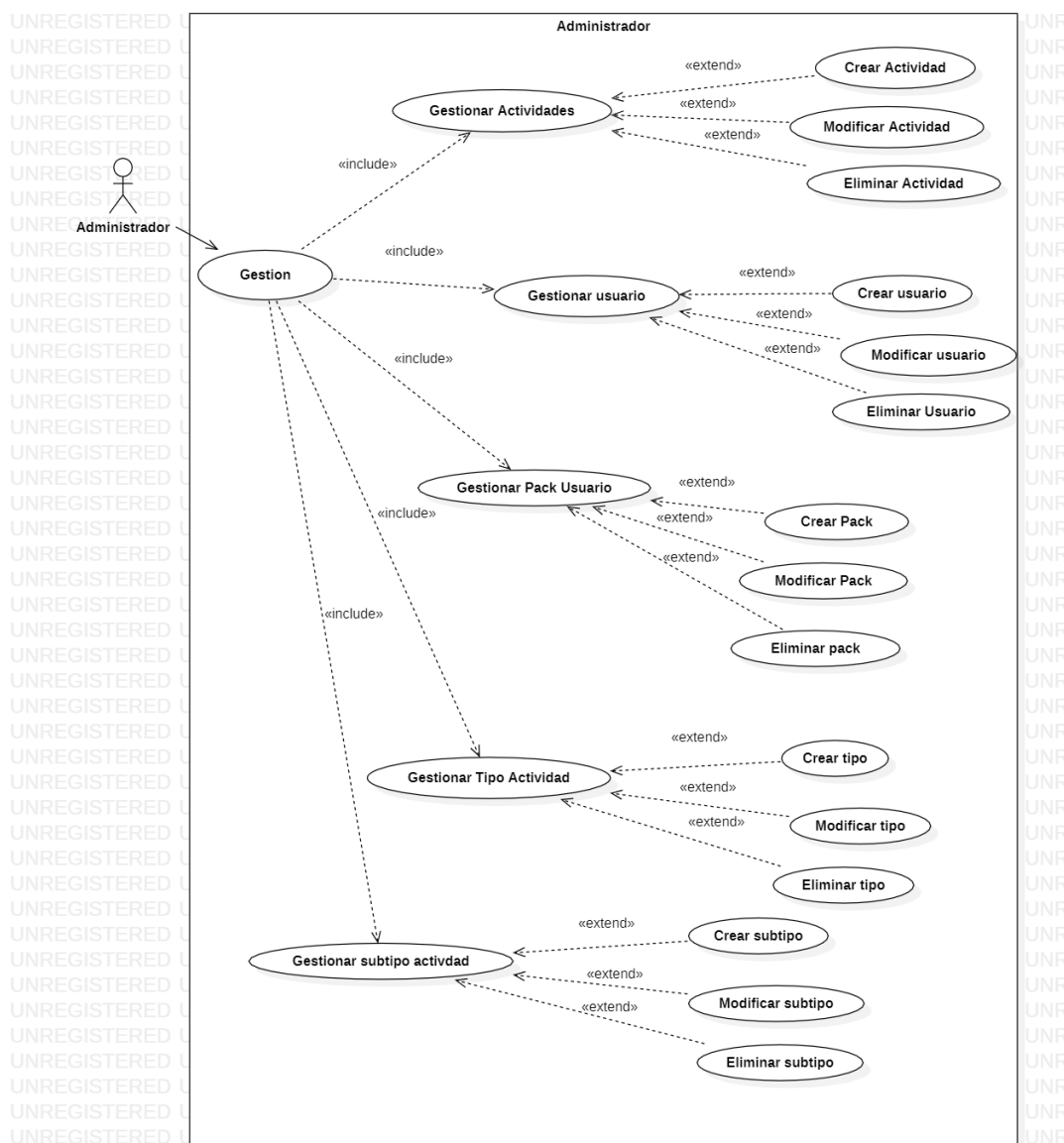
Estética: Específicamente, los colores deben de ser algo fríos, con una estética seria, sin recargar la estética.

Sin nada más que añadir las partes, se establecen como bases del proyecto la información detallada anteriormente. En el caso de querer añadir o retirar algún tipo de requisito, este documento justifica cualquier modificación en el presupuesto del proyecto.

DIAGRAMA DE CASOS DE USO

En este apartado se mostrará gráficamente y de la manera más intuitiva posible, las relaciones de los diferentes usuarios frente a la aplicación. Qué cosas podrán hacer y qué no, y la manera en la que los roles interactúan entre ellos para la realización de las diferentes tareas que pueden surgir a lo largo del uso de la aplicación solicitada.

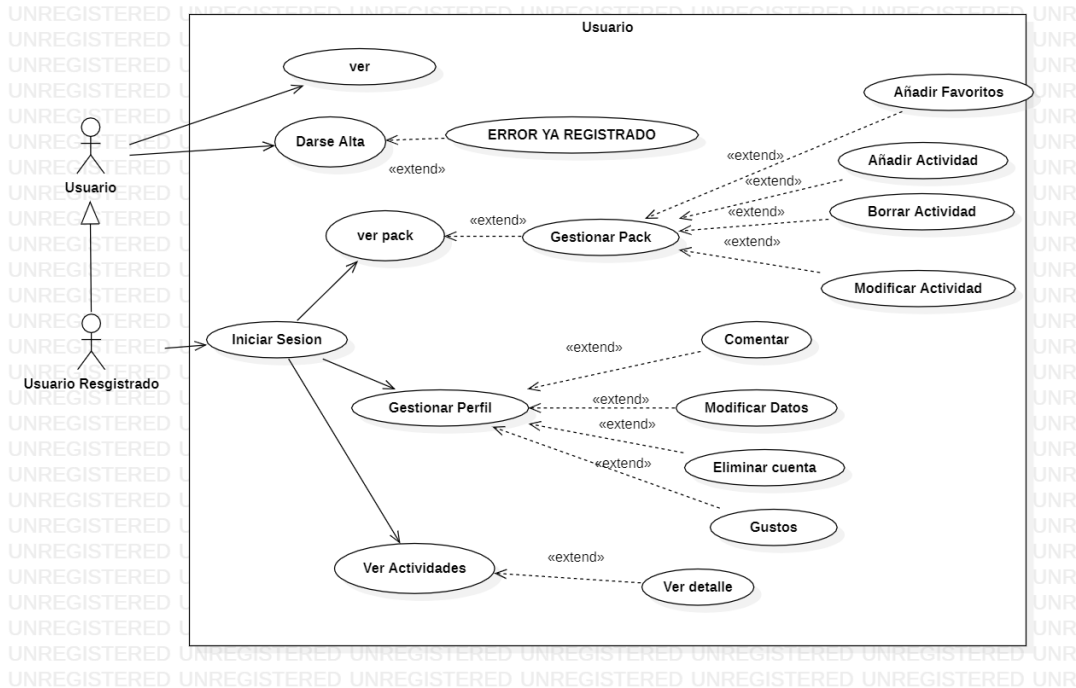
Atendiendo a los objetivos descritos en los ERS anteriores, los diagramas de casos de uso quedarían de la siguiente manera:



En cuanto a las tareas realizadas por el administrador (recordemos, un tipo de usuario registrado), constarán las de gestionar datos concretos de la aplicación, tanto de manera autónoma como de apoyo a un cliente que solicita ayuda. A saber:

1. **Gestión de Actividades:** Podrá crear, modificar y eliminar actividades que se oferten en la aplicación.

2. **Gestión Usuarios:** podrá crear usuarios por sí mismo, pero también podrá modificar perfiles registrados, como también eliminarlos si es necesario.
3. **Gestión Tipos:** En este apartado, el administrador podrá añadir, modificar o eliminar tipos de actividades, un dato importante para permitir la clasificación y filtrado de éstas.
4. **Gestión Subtipos:** en la misma línea que la anterior, pero en el caso de los subtipos, componente de las distintas actividades.



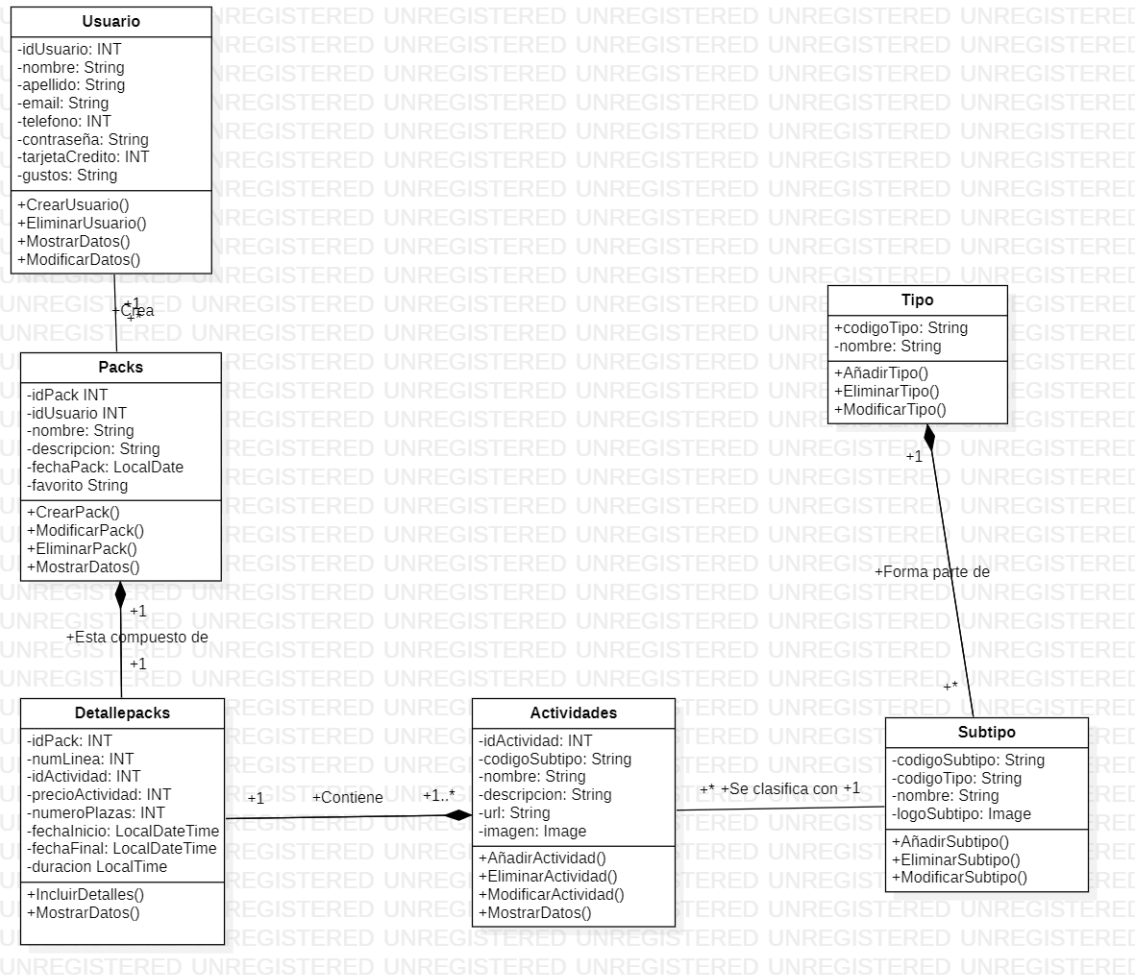
En cuanto a los casos de uso de los clientes, tenemos:

1. Un **cliente no registrado** solo podrá explorar la aplicación, pero no podrá crear packs ni gestionarlos, además de no poder gestionar su perfil.
2. Sin embargo, un **cliente registrado**, además de poder explorar, podrá en el caso que quiera:
 - a. **Gestionar Packs:** Pueden añadir actividades, borrarlas o modificar actividades (elegir otros días, distintas personas...). También pueden añadir el pack como favorito, en el caso de que quisiera volver a repetirlo o compartirlo.
 - b. **Gestionar Perfil:** puede comentar, modificar sus datos personales, añadir gustos personales y eliminar la cuenta si lo desea.
 - c. En el caso de que quiera ver las actividades que ha contratado, podrá verlas en detalle.

Para todo ello, el usuario registrado deberá iniciar sesión previamente.

DIAGRAMA DE CLASES DE LA APLICACIÓN

En este caso se mostrarán las clases de objetos que intervendrán en nuestra aplicación, y como interactúan entre sí. Atendiendo a los ERS, el diagrama de clases quedaría de esta manera¹:



El diagrama de clases está relacionado con la propia base de datos, asumiéndose la mayoría de los atributos como los utilizados en ésta. Los métodos por su parte están también estrechamente relacionados con la BD, permitiendo la aplicación añadir, eliminar o modificar datos para ser incluidos directamente en ella. Por último, se establece una función, que es la de poder mostrar datos concretos en la propia aplicación de manera discrecional según el tipo de usuario y ventana de la aplicación.

¹ Obviando getters y setters (añadir u obtener valores de parámetros de clases), los cuales asumimos como obligatorios, y el mostrar conjuntos de datos concretos de cada clase, los cuales se agrupan conceptualmente en el método MostrarDatos ().