

bonÀrea | IT

HackEPS 7ª edición

Nov. 23

bonÀrea ROCC

Recorrido Óptimo para Completar la Compra

THE FUTURE IS ALREADY HERE

it.bonarea.com

Minimización del tiempo destinado a realizar la compra y
visualización del recorrido

ÍNDICE

1. Qué es bonÀrea-IT.....	2
2. Qué necesidad tenemos y qué reto se plantea.....	3
2.1. ¿A quién le sobra tiempo para hacer la compra?	3
2.2. ¿En qué consiste el reto?	3
a. Calcular la ruta más rápida para completar la lista de la compra	3
b. Modificar la interfaz gráfica para incorporar nuevas funcionalidades	4
3. Datos.....	6
4. Restricciones	8
5. Interfaz gráfica	11
5.1. Cómo utilizar la interfaz gráfica	11
5.2. Estructura y código de la interfaz	13
5.3. Tareas de la interfaz gráfica	14
6. Formato de entrega	17
6.1. Código.....	17
6.2. Interfaz gráfica.....	17
6.3. Output.....	17
6.4. Presentación del proyecto	19
7. Evaluación	20
8. Premio	21

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

1. Qué es bonÀrea-IT

¿Sabíais que dentro de bonÀrea siempre se ha tenido la tecnología presente en todo el círculo productivo? Empezando con la digitalización del sector primario, pasando por las actividades industriales y finalmente con la distribución de los productos. Todo esto es gracias a un equipo técnico cualificado distribuido en diferentes áreas dentro de la empresa: bonÀrea-IT.



En [bonÀrea | IT](#) podréis conocer un poco más qué hacemos dentro de bonÀrea-IT, las distintas áreas que lo forman o parte del equipo que trabaja diariamente, con la explicación de algunas de sus experiencias y opiniones recogidas en el vídeo [Conoce bonÀrea IT](#).

Minimización del tiempo destinado a realizar la compra y
visualización del recorrido

2. Qué necesidad tenemos y qué reto se plantea

2.1. ¿A quién le sobra tiempo para hacer la compra?

Nuestro día a día (clases, prácticas, extraescolares, salir con los/as amigos/as, eventos deportivos importantes, etc.) nos deja poco tiempo libre para tareas necesarias como hacer la compra. Considerando esta realidad, sería fantástico que hubiese una aplicación que, dada una lista de productos que nos hacen falta, nos ayudase a realizar la compra de la forma más rápida posible.

Teniendo esto en cuenta, hemos decidido ‘proponer’ un reto cuya finalidad es desarrollar dicha aplicación que ayude al/a la cliente/a a realizar la compra de forma rápida y, a la vez, permita a la tienda visualizar el flujo de los/as clientes/as que se encuentran en la tienda en ese momento.

2.2. ¿En qué consiste el reto?

El reto para completar la lista de la compra de cada cliente/a de la forma más ágil y rápida consiste en la realización de dos partes:

- Crear un optimizador de rutas para calcular (y escribir en un fichero) la ruta óptima para cada cliente/a, considerando óptimo como el recorrido que requiere del menor tiempo para completar correctamente la lista de la compra.
Esta optimización debe hacerse desde el punto de vista de cada cliente/a, es decir, de conseguir el mínimo tiempo que necesita cada cliente/a sin verse alterado por el conjunto de tiempo que requieren todos los/as clientes/as a rellenar su cesta. Eso sí, siempre cumpliendo con todas las restricciones (ver [Restricciones](#)).
- Ampliar y mejorar una interfaz gráfica ya existente del mapa de una tienda, para poder visualizar los trayectos (óptimos) de sus clientes/as. Y, además, con el extra que os ayudará a verificar las rutas calculadas por el optimizador del punto anterior.

a. Calcular la ruta más rápida para completar la lista de la compra

La parte del optimizador de la ruta corresponde al **60%** del total final y está dividida en los siguientes dos niveles, cada uno independiente y con una puntuación del **30%** respecto a la nota final.

1. Nivel 1: Cálculo independiente por cliente/a (30% de la puntuación total)

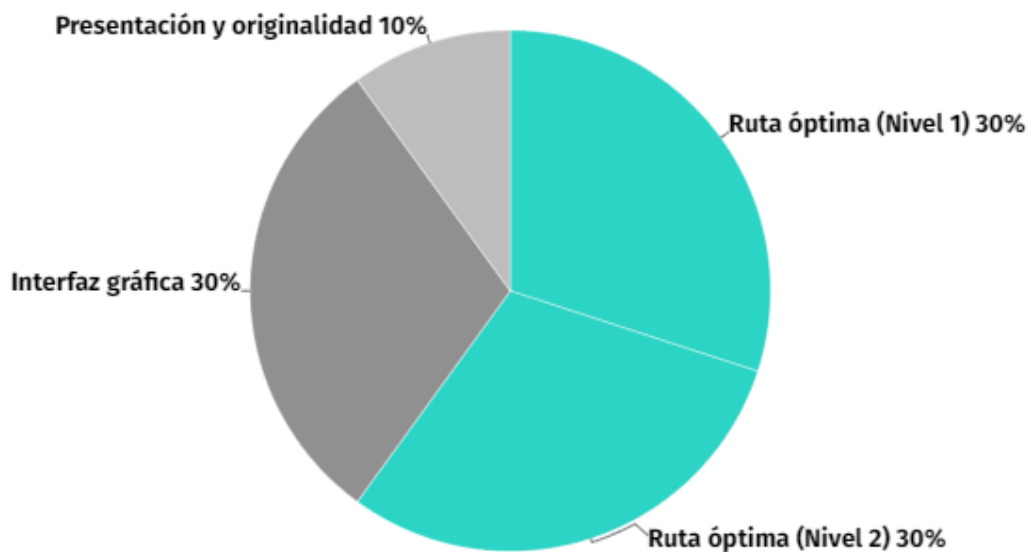
Calcular la ruta más rápida, es decir, con el menor tiempo posible, para cada persona, como si no hubiese otros/as clientes/as en la tienda que puedan interferir en el recorrido de la persona.

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

2. **Nivel 2:** Cálculo considerando toda la clientela (**30%** de la puntuación total)

Calcular la ruta más rápida, es decir, con el menor tiempo posible, para cada persona, considerando otros/as posibles clientes/as en la tienda que puedan interferir en el recorrido de cada persona.

En resumen, la puntuación por niveles de la parte del optimizador es la siguiente:



b. Modificar la interfaz gráfica para incorporar nuevas funcionalidades

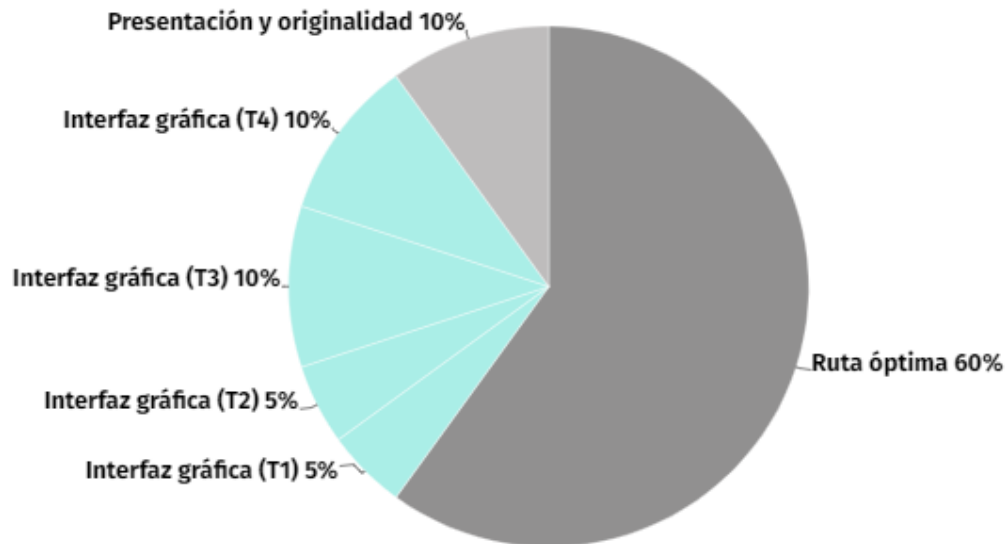
La segunda parte del reto consiste en ampliar la interfaz gráfica entregada y equivaldrá a un **30%** sobre la nota final, dividida en las siguientes cuatro tareas, cada una con su puntuación:

- T1. Remarcar la casilla del producto durante el tiempo de *picking* (**5%**)
- T2. Diferenciar la parte prevista y la parte recorrida de la ruta (**5%**)
- T3. Mostrar la ruta de los/as clientes/as en una tabla (**10%**)
- T4. Añadir otras funcionalidades a la interfaz gráfica (**10%**)

Cada una de estas tareas está explicada con más detalle en el apartado [Tareas de la interfaz gráfica](#).

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

El resumen gráfico de la puntuación de la parte de la interfaz gráfica es el siguiente:



Minimización del tiempo destinado a realizar la compra y
visualización del recorrido

3.Datos

Los datos que os proporcionamos vienen separados en cuatro conjuntos de datos descritos mediante las siguientes características:

- **<planogram_table.csv>**: conjunto de datos que incluye una descripción de las coordenadas (X, Y) de cada elemento de la tienda: almacén, zona de paso, estanterías con cada producto, entre otras.
 - **x**: número que indica la coordenada X de cada elemento del mapa de la tienda.
 - **y**: número que indica la coordenada Y de cada elemento del mapa de la tienda.
 - **picking_x**: número que indica la coordenada X por donde se coge el producto que corresponda (*picking*).
 - **picking_y**: número que indica la coordenada Y por donde se coge el producto que corresponda.
 - **description**: descripción del elemento al cual pertenecen las coordenadas.
- **<hackathon_customers_properties.csv>**: conjunto de datos con la lista de los/as clientes/as de la tienda y sus propiedades de desplazamiento y compra.
 - **customer_id**: número identificativo y único de cada cliente/a.
 - **step_seconds**: número de segundos que se tiene que permanecer en una casilla antes de desplazarse a la siguiente.
 - **picking_offset**: tiempo que tarda cada cliente/a en coger un producto.
- **<hackathon_articles_picking_time.csv>**: conjunto de datos con la descripción de los productos de la tienda y el tiempo que se tarda en coger cada unidad.
 - **article_id**: dato categórico que indica el código de cada producto disponible en la tienda.
 - **article_name**: nombre de cada producto disponible en la tienda.
 - **first_pick**: tiempo que se tarda en coger la primera unidad de un producto.
 - **second_pick**: tiempo que se tarda en coger la segunda unidad de un producto.
 - **third_pick**: tiempo que se tarda en coger la tercera unidad de un producto.
 - **fourth_pick**: tiempo que se tarda en coger la cuarta unidad de un producto.
 - **fifth_more_pick**: tiempo que se tarda en coger la quinta unidad (o más) de un producto.

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

- [<hackathon_tickets.csv>](#): conjunto de datos con la lista de tíquets de compra de cada cliente/a con los productos a comprar.
 - **enter_date_time**: fecha y hora mínima de entrada de la persona en la tienda.
 - **customer_id**: número identificativo y único de cada cliente/a.
 - **article_id**: dato categórico que indica el código de cada producto comprado.
 - **quantity**: número que indica las unidades que se cogen de cada producto.
 - **ticket_id**: dato categórico que indica el código de cada tíquet.

Además de estos cuatro conjuntos de datos, también os proporcionaremos el documento [<planogram_graphic.xlsx>](#), que corresponde al planograma de la tienda con sus coordenadas. Lo tendréis para que podáis tener una referencia más visual de las casillas y de las localizaciones de los productos (coordenadas (X, Y)). Y el documento [<output_test.csv>](#), que os servirá como archivo de prueba para la interfaz gráfica (ver [Cómo utilizar la interfaz gráfica](#)).

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

4. Restricciones

Restricciones para tener en cuenta para poder puntuar en la parte del optimizador de rutas (**60%** de la puntuación total). **El incumplimiento de cualquiera de ellas (excepto la última, que sólo se considera en el Nivel 2), implicará una calificación igual a 0 de esta parte.** Evaluaremos vuestras restricciones con un código verificador.



Nivel 1



Nivel 2



Solo se permiten movimientos horizontales y verticales.



El salto o paso de una casilla hacia la siguiente siempre tiene que ser entre dos casillas contiguas.



Sólo se permite el paso por las casillas (X, Y) de la tienda (definidas en *planogram_table.csv*) la descripción de las cuales contiene la palabra *paso* (*paso*, *paso-entrada*, *paso-salida*).



Las casillas *paso-salida* (ver *planogram_table.csv* o *planogram_graphic.xlsx*) son exclusivas para salir, no pueden ser utilizadas de paso en mitad de la ejecución de un tíquet. En otras palabras, en estas casillas sólo se permite un movimiento vertical y hacia abajo.



Se tienen que completar todos los tíquets (listas de la compra, ver *hackathon_tickets.csv*) que se han solicitado. Es decir, todos los productos de cada tíquet deben recogerse correctamente.



En el fichero *output.csv* se debe tener información de la posición de cada cliente/a en cada instante de tiempo.



Cada persona debe moverse según el tiempo que tarda a recorrer una casilla, el cual es diferente para cada una de ellas y se encuentra al fichero *hackathon_customers_properties.csv*.



La primera información de cada tíquet de cada persona debe corresponder a una casilla de entrada *paso-entrada* (ver *planogram_table.csv* o *planogram_graphic.xlsx*). Indicando así que la persona ha entrado a la tienda.

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

Veamos un ejemplo:

Al fichero *hackathon_customers_properties.csv* tenemos

<i>customer_id</i>	<i>step_seconds</i>
c0	5

Entonces el tiempo de estancia a la casilla de entrada será de mínimo 5 segundos. Se podría dar el caso que haya otra persona que bloquee la siguiente casilla destino, entonces el/la cliente/a c0 se tendría que esperar en la casilla de entrada (casilla paso-entrada) el tiempo extra que se requiriese.



La última información de cada tíquet de cada persona debe corresponder a una casilla de salida *paso-salida* (ver *planogram_table.csv* o *planogram_graphic.xlsx*). Indicando así que la persona ha salido de la tienda.



Los productos se tienen que coger desde las coordenadas especificadas (ver *planogram_table.csv*).



Para cada tíquet, el tiempo de recogida de un producto a $t=N$ se refiere a cuando, de manera consecutiva y sin desplazamiento, se coge la unidad número N del mismo producto. En caso de que hubiera desplazamiento, el tiempo de recogida del siguiente producto se volvería a posicionar a $t=1$.

Os recordamos que el procedimiento de recogida de productos empieza en el mismo instante en el cual se llega a la casilla de recogida (X, Y).

Veamos un ejemplo:

Si un/a cliente/a c0 quiere coger $N=6$ unidades del artículo a8007 y el fichero *hackathon_articles_picking_time.csv* nos indica

<i>article_id</i>	<i>article_name</i>	<i>first_pick</i>	<i>second_pick</i>	<i>third_pick</i>	<i>fourth_pick</i>	<i>fifth_more_pick</i>
a8007	Harina de trigo bonÀrea 1K	13	12	10	9	7

Y al fichero *hackathon_customers_properties.csv* tenemos

<i>customer_id</i>	<i>step_seconds</i>	<i>picking_offset</i>
c0	5	1

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

Entonces el tiempo de recogida de este producto sería

$$\langle \text{first_pick} \rangle + \langle \text{second_pick} \rangle + \langle \text{third_pick} \rangle + \langle \text{fourth_pick} \rangle + \langle N-5+1 \rangle * \langle \text{fifth_more_pick} \rangle + \langle N \rangle * \langle \text{picking_offset} \rangle = 13+12+10+9+(6-5+1)*7+6*1 = 64s$$

Este/a cliente/a estará 64 segundos a la casilla A cogiendo el artículo a8007, con el campo $\langle \text{is_picking} \rangle = 1$, y después iniciará su paso a la siguiente casilla vacía, con el campo $\langle \text{is_picking} \rangle = 0$. Por lo tanto, estará un total de $64 + \langle \text{step_seconds} \rangle = 64+5 = 69$ segundos en la casilla A.



Cada persona para cada tíquet debe entrar a la tienda a la hora especificada en el fichero *hackathon_tickets.csv*. Si se escoge el Nivel 2, una persona puede entrar más tarde de la hora especificada si se da el caso que la casilla de entrada (casilla *paso-entrada*) está ocupada por otra persona, ya que dos personas no pueden ocupar la misma casilla en el mismo instante de tiempo (ver última restricción).



A la casilla de pago *paso-salida* (ver *planogram_table.csv* o *planogram_graphic.xlsx*) permanecerá un tiempo extra de **5 segundos** por cada unidad de artículo a pagar. Es decir, si una persona sale de la tienda habiendo comprado N unidades en total, permanecerá en la casilla $5*N$ + tiempo de desplazamiento de la casilla necesario para ir a la siguiente.

Veamos un ejemplo:

Al fichero *hackathon_customers_properties.csv* tenemos

customer_id	step_seconds
c0	4

Entonces el tiempo de pago de una compra total de 6 unidades a8007 en la caja sería $5*\langle N \rangle = 5*6 = 30s$.

Este/a cliente/a estará 30 segundos pagando los artículos, y después iniciará su paso para salir de la tienda. Por lo tanto, estará un total de $30 + \langle \text{step_seconds} \rangle = 30+4 = 34$ segundos en la casilla *paso-salida*.



Se debe entregar un formato output determinado de la ruta óptima que hayáis encontrado (ver [Formato de entrega](#)).



(Sólo en el caso de que escogéis el Nivel 2 del optimizador de rutas) Una persona no puede situarse en una casilla que ya esté ocupada por otra persona en el mismo instante de tiempo. En otras palabras, a cada coordenada (X, Y) solo cabe una única persona en un instante de tiempo determinado.

Minimización del tiempo destinado a realizar la compra y
visualización del recorrido

5. Interfaz gráfica

5.1. Cómo utilizar la interfaz gráfica

Para poder disponer de la interfaz gráfica, deberéis descargar la carpeta **UI** facilitada por el equipo de bonÀrea IT por el canal de Slack, una vez descargada, deberéis abrir el [index.html](#) en el navegador situado en la carpeta *Interfaz-gráfica* y en el *path* donde se haya instalado el proyecto. Por ejemplo:

[C:/Users/\[user.name\]/\[my.projects\]/ui/index.html](C:/Users/[user.name]/[my.projects]/ui/index.html)

Una vez os cargue os saldrá la siguiente imagen:



Tened en cuenta que el eje de coordenadas empieza en la esquina superior izquierda, tal como se aprecia en el plano, y ésta va aumentando una unidad por casilla, siendo la **X** el eje horizontal y la **Y** el eje vertical y yendo en este orden (X, Y).

Con tal de usar esta herramienta, primero deberéis ajustar la velocidad con la que os gustaría ver la simulación, que por defecto estará en 1: el nivel 1 corresponde a la velocidad *normal*, el nivel 2 es la velocidad multiplicada por 10, el nivel 3 multiplicada por 100 y el nivel 4 multiplicada por 1000 (que es el nivel máximo). Luego deberéis cargar el archivo, si lo hacéis al revés, la velocidad sólo se aplicará para algunos/as clientes/as. El archivo que deberéis cargar deberá estar en formato CSV tal como se muestra en el ejemplo y tal como se especifica en el apartado de entrega del output (ver [Output](#)).

Minimización del tiempo destinado a realizar la compra y
visualización del recorrido

ROCC: Recorrido Óptimo para Completar la Compra.

Carga tu archivo CSV aquí: No se ha seleccionado ningún archivo



Velocidad Actual: 1



Plano de la tienda.

Para poder probarlo, os podréis descargar un archivo de prueba, *output_test.csv*, adjuntado con el resto de la documentación facilitada en el canal de Slack, pero para utilizarlo con vuestros datos deberéis generar primero vuestro propio archivo *output_n1.csv* o *output_n2.csv*.

Esta herramienta también os ayudará a visualizar si las rutas que generéis colisionan entre ellas. Recordad que, de hecho, las rutas no pueden colisionar entre ellas en el Nivel 2 del optimizador.

ROCC: Recorrido Óptimo para Completar la Compra.

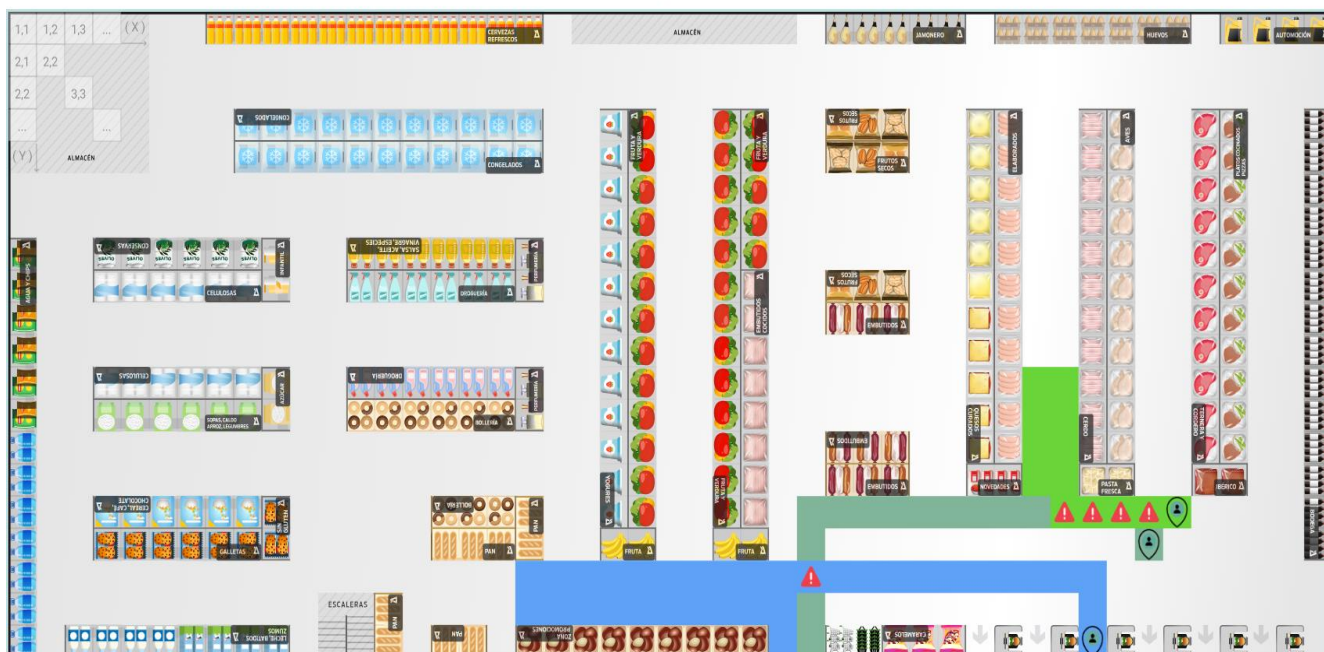
Carga tu archivo CSV aquí: collisionRoute.csv



Velocidad Actual: 1



Plano de la tienda.



Minimización del tiempo destinado a realizar la compra y visualización del recorrido

5.2. Estructura y código de la interfaz

En la carpeta **UI** del proyecto encontraréis otra carpeta con las imágenes necesarias para la visualización de la herramienta, un archivo **index.html**, un **style.css** y un **script.js**. Todos estos archivos los podéis modificar o añadir nuevos con la intención de mejorar la herramienta visual para obtener mejor puntuación, tal y como se especifica en el apartado [Modificar la interfaz gráfica para incorporar nuevas funcionalidades](#).

En la carpeta **img** encontraréis las siguientes imágenes:

- **planogram_image.svg**: mapa de la tienda.
- **marker_client.png**: icono para marcar la posición de un/a cliente/a.
- **red_alert.svg**: icono para marcar las colisiones.

En el archivo de **script.js** encontraréis la siguiente información:

Constantes

- **ctxIconClient**: variable que permite a *canvas* dibujar el icono del/de la cliente/a.
- **ctxIconCollition**: variable que permite a *canvas* dibujar el icono de la colisión.
- **ctxSquare**: variable que permite a *canvas* dibujar el cuadrado de la ruta.
- **imgCollition**: imagen para simular la colisión de los/as clientes/as.
- **DIM**: dimensión de la casilla (40).

Variables

- **tickets**: mapa del número de tíquets con su valor.
- **customerColor**: mapa con el número de tíquet y el color asignado.
- **customerTickets**: mapa con el número de tíquet y el color asignado.
- **locationsShared**: listado con todas las ubicaciones que dos clientes/as o más comparten.
- **locationsCollition**: listado con todas las ubicaciones que dos o más clientes/as chocan, es decir, que comparten posición en el mismo segundo.
- **locationsTotal**: listado de las ubicaciones, segundos y número de tíquet.

Funciones

- **readFile()**: función que lee el *output.csv* que se carga desde la web.
- **getDataOfFile()**: función que recorre cada línea del *output.csv* (excepto la primera línea), y guarda los valores en las variables *customer_id*, *ticket_id*, *x*, *y*, *picking* y *x_y_date_time*.
- **calcWaypoints(locations)**: función que crea un *array* con las localizaciones y tiempo para cada cliente/a.

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

- **calculateFirstClientSec():** función asíncrona que calcula para cada cliente/a el segundo en el que entra en la tienda y todas las coordenadas de su ruta.
- **drawRouteAfterSeconds():** función que espera los segundos calculados en la función anterior y espera esos segundos antes de llamar a la función de dibujar ruta.
- **drawRoute():** función que se encarga de pintar cada casilla del color del/de la cliente/a en el momento que este/a se desplaza a ella.
- **clearRoute():** función que borra la estela del/de la cliente/a una vez ha salido de la tienda.
- **drawSquare():** función que pinta de color toda la superficie de la casilla.
- **generateColor():** función que genera un color aleatorio para cada cliente/a, reservando el blanco (ya que no se vería al ser el fondo del mismo color).
- **sleep():** función que pausa N milisegundos antes de hacer el próximo proceso.
- **modifySpeed():** función que aumenta o reduce la velocidad según lo que recibe del HTML y la modifica también en la visualización.
- **epochConverter():** función que convierte la fecha y hora leídas desde el *output_n1.csv/output_n2.csv* en *epochtime* para poder trabajarlas como un conjunto.
- **sortRouteByTime():** función que ordena el *array* de ubicaciones y lo devuelve ordenado.
- **caculateOpenDate():** función que se encarga de leer el día del *output_n1.csv/output_n2.csv* configurando la hora establecida a 9:00AM (que es la hora que se abre la tienda) y guardándolas como un conjunto en *epochtime*.
- **drawLocationsCollition(x,y):** función que inserta la imagen de la colisión, según los puntos *x* e *y*.
- **getSharedAndCollitionLocations():** función que calcula previamente las coordenadas donde se cruzan los recorridos de cada cliente/a y los guarda en un mapa de valores de solapamientos o de colisiones según el caso.

5.3. Tareas de la interfaz gráfica

Con la intención de evaluar los conocimientos sobre el manejo de una interfaz gráfica, os hemos propuesto unas mejoras que esperamos que implementéis en la interfaz que ponemos a vuestra disposición. Las tareas para realizar estas mejoras constan de cuatro puntos mencionados en un apartado anterior (ver [Modificar la interfaz gráfica para incorporar nuevas funcionalidades](#)) y se evaluarán según los porcentajes expresados en ese mismo apartado. No es necesario que desarrolléis las tareas en el orden en que se numeran, pero sí las debéis implementar según las indicaciones que se os presentan a continuación:

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

T1. Remarcar la casilla del producto durante el tiempo de picking.

Se espera que uséis algún sistema para destacar (ya sea con color, sombreado, icono, marco, etc.) la casilla donde está localizado el artículo que el/la cliente/a está adquiriendo en aquel momento. Tened en cuenta que el icono de cada artículo está adherido al mapa, de forma que si optáis por una opción que repinte la casilla esta también ocultará el icono del artículo.

T2. Diferenciar la parte prevista y la parte recorrida de la ruta.

Identificando en todo momento las diferentes rutas de los/as clientes/as, es decir, que no se confundan las rutas de los/as diferentes clientes/as, debéis plasmar en el mapa, de forma visual y diferenciada, el recorrido que tiene previsto hacer el/la cliente/a y la parte de la ruta ya efectuada. Lo podéis hacer valiéndoos de sombreados, interlineados, remarcados, iconos o cualquier otra solución que se os ocurra, tanto para la ruta recorrida como para la prevista.

T3. Mostrar la ruta de los/as clientes/as en una tabla.

En este apartado se pretende conseguir un resumen de manera visual de toda la información que se tiene sobre las rutas y el estado de estas. Esta tabla se debe mostrar en la misma página del visualizador, justo debajo del mapa de la tienda, con la siguiente información:

- **Estado Ruta:** Completado / En ruta / En espera.
 - Un apartado que complemente el estado de la ruta de manera visual (semáforo).
- **N.º Cliente/a:** número identificador de cliente/a.
- **Hora Entrada:** hora de entrada del/de la cliente/a en la tienda.
- **Hora Salida:** hora de salida del/de la cliente/a en la tienda.
- **Duración:** tiempo que el/la cliente/a ha pasado o pasará en la tienda.
- **N.º Ticket:** número identificador del tíquet.
- **Cantidad de Artículos:** cantidad de artículos que hay en el tíquet.

Aparte de esta información se podría añadir cualquier otra que consideréis relevante. La tabla debería quedar parecida a esta:

Estado Ruta	Nº Cliente/a	Hora Entrada	Hora Salida	Duración	Nº Ticket	Cantidad Artículos
 Completado	0001	09:35:25	09:45:26	10 min. 1s	T20354	23
 En ruta	0002	17:35:25	17:45:35	10 min. 10s	T20355	5
 En ruta	0003	17:45:00	18:15:00	30 min	T20388	40
 En espera	0023	19:25:00	19:45:10	20 min 10s	T20580	12

Minimización del tiempo destinado a realizar la compra y
visualización del recorrido

T4. Añadir otras funcionalidades a la interfaz gráfica.

Este es el apartado más libre, se espera que plasméis vuestros conocimientos y dominios sobre la web para transformar la estética de la página, incluyendo los componentes de ésta. Para ello se podrá modificar cualquiera de los archivos proporcionados, así como añadir otros archivos nuevos, si lo consideráis necesario. Aparte de los aspectos visuales podéis modificar o remplazar cualquier parte de la interfaz que creáis conveniente o cualquier mejora que se os ocurra. Recordad que este apartado se puntuará sobre la estética global, los cambios efectuados y la originalidad de éstos.

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

6. Formato de entrega

Deberéis crear una carpeta **zip** con el nombre de vuestro equipo (**nombre_del_equipo.zip**) y subirla al **grupo privado** de *Slack* de vuestro equipo (no lo subáis en el grupo de archivos, ni en el general, ya que estaríais compartiendo vuestro trabajo con los demás grupos), dentro de esta carpeta deberéis adjuntar los ficheros que queráis entregar, de la siguiente manera:

- **UI/***
- **OPTIMIZER/output/output_n1.csv, output_n2.csv**
- **OPTIMIZER/code/***
- **PRESENTATION/* (fichero opcional)**

6.1. Código

El código del optimizador se puede entregar con el lenguaje de programación que elegís vosotros (Python, C, Java, etc).

6.2. Interfaz gráfica

Para la entrega de la interfaz gráfica se deberá entregar la carpeta que se facilitó al principio (en la documentación subida en *Slack*), incluyendo todos los archivos del proyecto, hayan sido modificados o no.

Se puede utilizar el lenguaje de programación que queráis, aun así, tened en cuenta que la base que os proporcionamos está desarrollada en JavaScript.

6.3. Output

Se deberán entregar dos ficheros **csv**, uno para cada nivel del optimizador, con los siguientes nombres: **output_n1.csv** para el Nivel 1 y **output_n2.csv** para el Nivel 2, las columnas que deben contener tienen que ser las siguientes, separadas por **punto y coma** y con el siguiente orden:

- **customer_id**: número identificativo y único de cada cliente/a.
- **ticket_id**: dato categórico que indica el código de cada producto comprado.
- **x**: número que indica la posición X de la casilla en un instante de tiempo.
- **y**: número que indica la posición Y de la casilla en un instante de tiempo.
- **picking**: valor binario que indica si ha cogido, 1, o no, 0, el producto.
- **x_y_date_time**: número que indica cada instante de tiempo, el primero está indicado al fichero *tickets.csv* y a los siguientes a este se le debe sumar 1 segundo por cada fila del fichero *output.csv*. El formato de esta columna debe cumplir las siguientes condiciones:

Minimización del tiempo destinado a realizar la compra y visualización del recorrido

- Indicar la fecha y la hora separadas por **tan sólo** un espacio.
- El formato de la fecha debe ser *aaaa-mm-dd* (año-mes-día separados por guion)
- El formato de la hora debe de ser *hh:mm:ss* (ejemplo 08:05:01).

Recordamos que cada fichero debe respetar el orden mencionado de las columnas y debe contener segundo a segundo el tiempo que permanece el/la cliente/a en la tienda sin que haya ningún salto en la continuidad de segundos. Además, no puede contener ningún tipo de valor nulo (*NaN*, *NULL*, *None*, *Undefined*, ‘’, espacios en blanco, etc). Abajo mostramos un ejemplo de output:

Output_ordenado.csv						
customer_id	ticket_id	x	y	picking	x_y_date_time	
0	t123456	29	20	0	2023-08-24 09:00:00	
0	t123456	29	19	0	2023-08-24 09:00:01	
0	t123456	28	19	0	2023-08-24 09:00:02	
0	t123456	27	19	0	2023-08-04 09:00:03	
0	t123456	26	19	0	2023-08-04 09:00:04	
0	t123456	25	19	0	2023-08-04 09:00:05	
0	t123456	24	19	0	2023-08-04 09:00:06	
0	t123456	23	19	0	2023-08-04 09:00:07	
0	t123456	22	19	0	2023-08-04 09:00:08	
0	t123456	21	19	0	2023-08-04 09:00:09	
0	t123456	20	19	0	2023-08-04 09:00:10	
1	t123457	29	20	0	2023-08-04 09:01:40	
1	t123457	29	20	0	2023-08-04 09:01:41	
1	t123457	29	19	0	2023-08-04 09:01:42	
1	t123457	29	19	0	2023-08-04 09:01:43	
1	t123457	29	18	0	2023-08-04 09:01:44	
1	t123457	29	18	0	2023-08-04 09:01:45	
2	t123458	29	20	0	2023-08-04 09:00:38	
[...]						

**Recomendamos que para trabajar con CSV tengáis cuidado al abrirlo en Excel, ya que el formato de los datetimes pueden verse con un formato no apropiado. De todas formas, podéis especificar el formato en las celdas de estos datetimes (aaaa-mm-dd hh:mm:ss) para una correcta visualización.*

Una herramienta alternativa para abrir el archivo puede ser un bloc de notas, por ejemplo.

Nota: Si a la hora de implementar las tareas de la interfaz gráfica necesitáis añadir alguna columna a la derecha del archivo (después de la del tiempo *x_y_date_time*), la podéis añadir siempre y cuando tratéis la columna en la interfaz gráfica.

Minimización del tiempo destinado a realizar la compra y
visualización del recorrido

6.4. Presentación del proyecto

Se deberá hacer una defensa del proyecto y los resultados obtenidos. Podéis ayudaros de un material de soporte tal como PowerPoint, Canva, Prezi, etc., que a su vez podéis entregar (si queréis) en la carpeta de **Presentation/***.

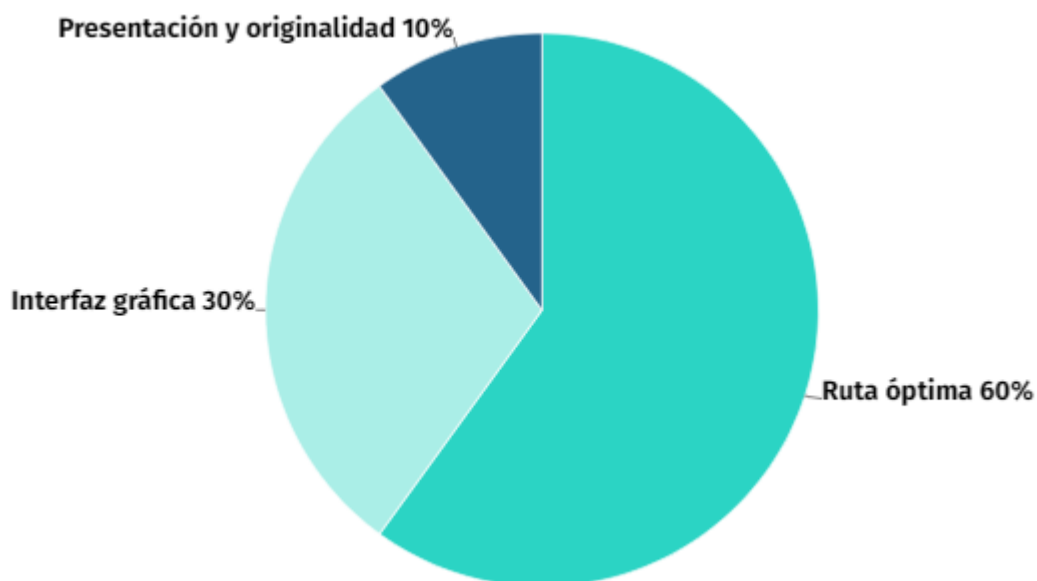
Minimización del tiempo destinado a realizar la compra y visualización del recorrido

7.Evaluación

En la evaluación se tendrán en cuenta distintos conceptos, entre ellos la calidad, los resultados, la programación, así como la originalidad y la presentación del proyecto.

La nota final será el resultado de las diferentes puntuaciones que se obtendrán de cada apartado valorado por los porcentajes presentados a continuación:

- **60% Ruta óptima:** ficheros *output_n1.csv* y *output_n2.csv* con la solución de la ruta óptima encontrada para cada nivel.
- **30% Interfaz gráfica (UI):** fichero con las tareas hechas para la interfaz gráfica.
- **10% Presentación** y defensa del proyecto y **originalidad** detrás de los resultados.



Minimización del tiempo destinado a realizar la compra y visualización del recorrido

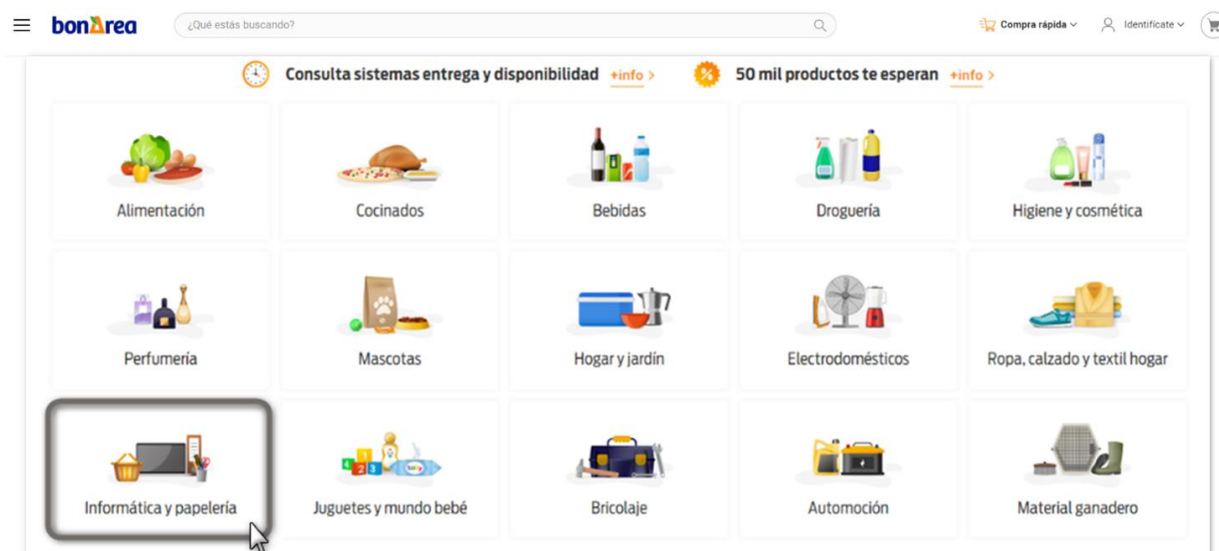
8. Premio

El premio consiste en un vale de **1.000€** para gastar en la tienda online de bonÀrea. Si el equipo ganador del premio desea algún dispositivo o componente tecnológico en concreto que no aparece en la tienda online, el departamento de compras de ingeniería se encargará de buscarlo.



Si queréis conocer todos los productos disponibles, echad un vistazo a la página web:

<https://www.bonarea-online.com/es>



**No olvidéis visitar la sección de productos de informática.*