

Entrega N° 1

Justificaciones de Diseño Iniciales

Diseño de contraseñas:

ValidadorDeContraseña: Clase principal para la verificación de las contraseñas.

Tiene una lista de condiciones de errores, que la contraseña a validar no debe cumplir ninguna, si cumple devuelve una lista con mensajes de error para posteriormente informarle al usuario los errores. Este es un componente stateless ya que almacena información aislada y sus entradas y salidas son iguales.

CondicionesErrorContraseña: Es una interfaz con los métodos cumpleError y mensajeDeError, cada uno depende de cual sea la condición a cumplir. Elegimos este diseño sabiendo que en un futuro se pueden agregar o quitar condiciones, pensando nuestro diseño más extensible, flexible y mantenible.

InfoUsuario: Esta clase está pensada para que nuestro diseño sea más extensible, ya que pueden surgir nuevas condiciones o se modifiquen algunas que requieran otros datos del usuario. Por esta razón las condiciones reciben un InfoUsuario para tomar de él los datos requeridos.

Logging: Es la clase encargada de crear una cuenta dependiendo del resultado que arroje el validador, si la lista está vacía crea la cuenta, de no ser así informará al usuario los errores correspondientes. Una vez creado el usuario lo agrega a la base de datos. Esta clase también tiene el método singIn ya que utiliza la DataBase.

—

Heladera: esta clase tiene como atributo una lista de viandas un ID para el json, así presentarlo en el mapa con sus coordenadas.

Coordenadas: Es una clase que tiene latitud y longitud. Elegimos este diseño ya que en un futuro puede calcular distancias o algún método el cual solamente coordenadas deba ser responsable. De esta manera obtenemos una mayor abstracción y reusabilidad.

—

Vulnerable: Esta clase conoce las heladeras ya que posteriormente deberemos implementar que el poder quitar una vianda para consumirla. Además tiene un documento opcional siendo este un DNI o LC o LE.

Colaboradores: Clase abstracta de las que heredan Jurídica y Humana. Usa Colaboración, en una lista para tener las contribuciones realizadas (entrega 2) y en una lista para contribuciones pendiente (implementado para la entrega 3), para contribuir polimórficamente los distintos tipos de contribuciones.

Colaboración: Interfaz de la que heredan todas las contribuciones que se pueden realizar.

—

Vianda: Esta clase es creada en el momento de donarVianda por parte de Humana.

Luego se instancia DonacionVianda para complementar los datos de Vianda y así cumplir la donación.

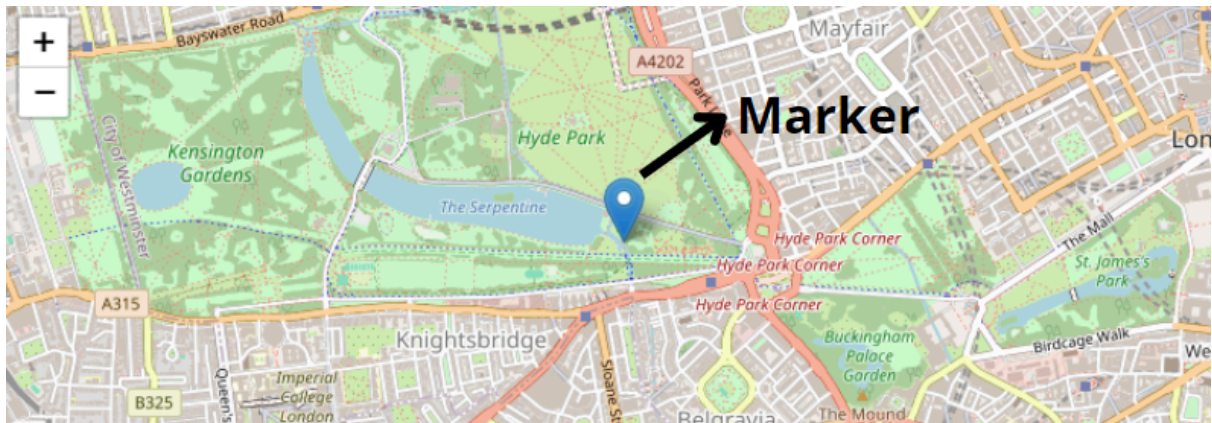
Implementación de UI

<https://www.figma.com/file/k3B0i3zl5M8LAHMwssHmiH/Frames?type=design&node-id=0%3A1&mode=design&t=H1d8OXqnlUFuN8WI-1>

Implementación del mapa donde se visualicen las heladeras.



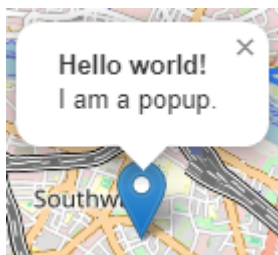
Leaflet es una librería con responsive design para JavaScript.



Marker: Un marker es un elemento de leaflet que permite agregar un marcador utilizando coordenadas (Latitud, longitud). De esta forma podríamos agregar cada una de las heladeras.

Además Leaflet es una buena alternativa ya que permite crear otro tipo de geometrías, áreas, etc.

Popups: Podemos agregar popups para reflejar la información de cada una de las heladeras.

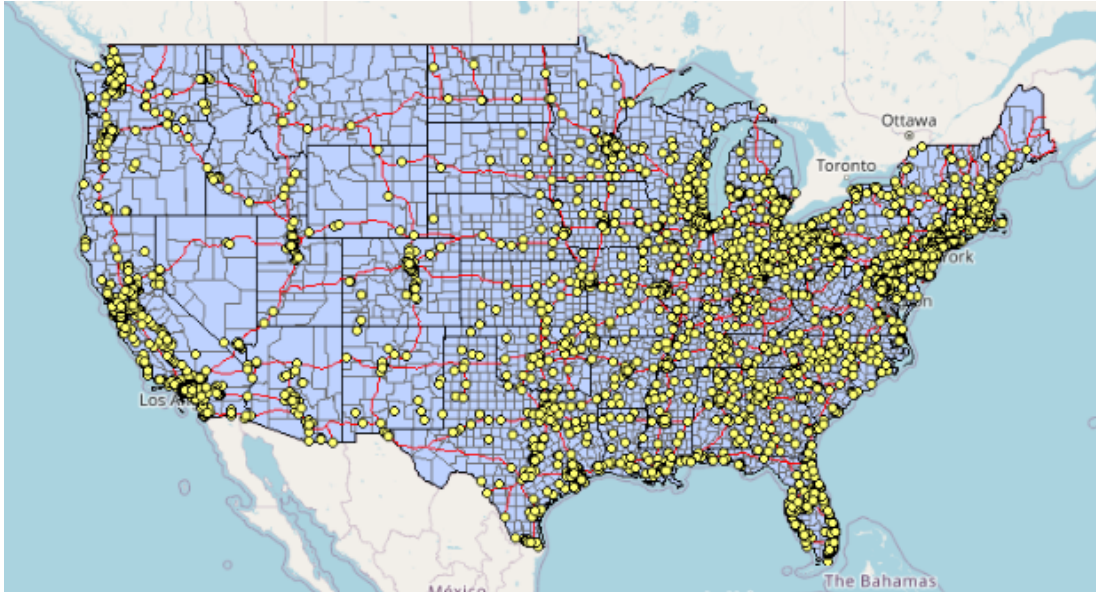




OpenLayers

Imagen representativa de Estados Unidos.

Es una librería con javascript para la creación de mapas. Tiene una buena documentación y ejemplos de aplicaciones de usuarios. Se pueden crear animaciones. Podemos visualizar cada una de las heladeras utilizando Points



Entrega N° 2

Personas en situación vulnerable

Modificamos la clase Vulnerable para tener constancia de la fechaRegistro.

Cambiamos el atributo menoresACargo por una lista de Vulnerable para llevar un control de las edades y registro propio de los Vulnerables menores de edad.

Modelamos una clase TarjetaPlastica que tiene un Vulnerable asociado. Esta tiene una lista de Uso para llevar una trazabilidad de los mismos, teniendo fecha y una heladera.

Con respecto a la colaboración que realiza Humana para registrar a la persona vulnerable y entregar la tarjeta modelamos una clase RegistroDePersonaVulnerable teniendo su fecha de contribución, la persona vulnerable a registrar, la cual recibirá la tarjeta y la propia persona Humana para llevar un registro de quien lo hizo.

Heladeras

Como el enunciado nos comentaba que las heladeras pueden sufrir desperfectos planteamos que en nuestro sistema hayan sensores que tengan una heladera asociada. Estos sensores van a actuar como receptores de la información que nos van a proveer los sensores físicos instalados en cada heladera.

- Sensor de temperatura: Se va a encargar de recibir una medición registrada por el sensor de temperatura de nuestra heladera física y la va a mandar al sistema, esta MediciónTemperatura contendrá la fecha de la medición y la última temperatura registrada. Una vez que el registro fue recepcionado por el sensor de nuestro sistema se va a cambiar la temperatura actual de la heladera asociada al sensor y va a ser esta la que verifique que todo esté en condiciones en base a su modelo que tiene la temperatura mínima y máxima por la cual la temperatura de la heladera debe oscilar. En caso de no estarlo, la misma deberá cambiar su estado a ENPELIGRO.
- Sensor de movimiento: Cuando el sensor de movimiento físico detecte algún movimiento extraño en la heladera enviará una alerta a nuestro sistema que será recepcionado por la entidad ReceptorDeAlerta que recibirá una AlertaMovimiento, que será una clase que asegurará la trazabilidad de las alertas que llegan a nuestro sistema proveyendonos de su fecha de lanzamiento y el sensor del cual nos provee.

Este receptor de alerta suponemos que también hará que se cambie el estado de la heladera de la cual fue lanzada aunque no especifica bien el enunciado.

Técnicos

La entidad Técnico será el encargado de solucionar los problemas de las heladeras que se encuentren en mantenimiento, respetando una cierta área de cobertura en la cual puedan trabajar. Para esto, implementamos una interfaz ICobertura que nos permite utilizar distintas formas en que se puede definir un área. Posiblemente se emplee el patrón strategy, en donde cada forma de representar el área de cobertura tendrá la responsabilidad de avisarnos si el técnico puede trabajar en una dirección o no.

Recomendación de puntos de colocación

La organización necesita recomendaciones de donde colocar sus heladeras en una determinada área. Por eso mismo en nuestro sistema, por medio del patrón adapter, vamos a adaptar los datos que nos llegan desde la API REST que tenemos a disposición y sabemos que la misma, aunque está todavía en desarrollo, por medio de una coordenada que va a contener una latitud y longitud, y un radio específico va a proveernos de dichas recomendaciones basándose en varios factores para determinarlas.

Reconocimiento a los colaboradores

Para el sistema de puntos pensamos que la mejor idea fue pensarlo como si fuera un sueldo mensual que refleja las contribuciones que hizo el colaborador durante ese mes. Para ello ideamos que mes a mes se realice un balance que contiene las contribuciones que le generaron puntos al colaborador (la clase Debe) y los canjes de objetos o servicios por la cantidad de puntos que este solicitaba (la clase Haber). Controlamos que los puntos no queden negativos haciendo que cuando se realice un canje verifique con el método puntosActuales(), los puntosAlCierre del anterior balance menos los puntos canjeados en la lista de Haber sean mayores a cero.

Carga masiva de colaboraciones

Bueno, empecemos. La carga masiva de colaboradores la realizamos a través de una biblioteca de Java llamada OpenCSV. Utilizamos la clase CSVReader para poder leer un archivo .CSV, que contiene la información de los colaboradores. Almacenamos cada uno de los registros del CSV (Filas) en un objeto ColaboradorCSV, que contiene toda la información. Luego podemos procesar todos estos datos almacenados en una lista y enviar a aquellos que no tengan una cuenta, un email.

Modificaciones en base a la entrega n°1:

- Mayor consistencia de datos en Dirección ya que hicimos que calle, localidad y provincia sean clases.
- Modelamos un Formulario para las inscripciones de las personas jurídicas y humanas, tiene lista de campo pudiendo ser de una opción o más o libre. Además tenemos un FormularioCompletado el cual tiene un Formulario y los camposCompletados.
- Agregamos un enumerado con los rubros que puede tener una persona jurídica. De esta forma evitamos problemas en la consistencia de datos y permitimos filtros futuros.
- No hicimos más uso de la clase abstracta colaboraciones, en su lugar para llevar la trazabilidad de las contribuciones que realizaron los usuarios creamos una clase distinta para cada una ellas. Cada una posee la fecha en la que se realizó la

colaboración, su actor correspondiente y algún dato más que haga referencia a la colaboración.

- Desistimos de la clase Colaboradores de la que heredaban Humana y Jurídica ya que su utilización no tenía ningún sentido dado a que era una clase que contenían los datos compartidos por ambas.