Per aconseguir el funcionament correcte del sistema s’ha dividit en dues parts: el back-end i el front-end.

Per una banda, el back-end consisteix en un servidor web basat en NodeJS i Express []. S’ha generat l’script essencial que forma part del codi “main” i s’han distribuït en diversos fitxers les funcions, per tal de que el codi sigui entenedor i manejable. Inicialment estava previst muntar el servidor web sobre una plataforma web anomenada Heroku []. Degut a que finalment no era adequada per termes econòmics, es va plantejar la solució d’Amazon Web Services [], la qual tampoc va resultar encertada. Finalment, per tal d’evitar complicacions i augmentar la complexitat del sistema en aquesta fase inicial, s’ha decidit fer funcionar el servidor de forma local a través d’una màquina virtual d’Ubuntu 20.04.2 LTS [] configurada amb tots els paquets i llibreries necessàries.

Per l’altra banda, el front-end consisteix en una aplicació multi-plataforma basada en el framework de Vue [], que utilitza JavaScript, juntament amb els frameworks d’Ionic [] i Capacitor [], que permeten la integració de l’aplicació web com a aplicació nativa tant en Android com en iOS i que permet l’ús del hardware (com el GPS). Aquesta aplicació està construïda amb diferents vistes mantenint sempre la barra de navegació inferior amb les pestanyes d’Inici, Cerca, Informació i Configuració. Degut a l’abast d’aquesta solució en fase inicial, l’aplicació no està disponible a la AppStore ni a la PlayStore.

En aquesta secció s’explica l’estructura final del sistema i com s’ha organitzat i aconseguit fer funcionar les dues parts que constitueixen i fan funcionar el sistema.

**SERVER**

Per la part del back-end del sistema s’ha organitzat en forma d’arbre, on el fitxer més important és el “main.js” que gestiona totes les peticions HTTP, crea i executa el servidor, i gestiona la comunicació amb la resta de fitxers JavaScript. Les funcions implementades en aquests fitxers envien les respostes a l’aplicació que les gestiona i mostra el resultat per pantalla, segons sigui necessari. A **la figura 7** es mostra l’estructura final de fitxers i directoris del servidor.

En primer lloc, els fitxers “SmartContract.js” i “tokenFunctions.js” inclouen totes les funcions necessàries i el document “.abi” per permetre la comunicació entre el servidor i el contracte que es troba a la xarxa de Ropsten. El directori “PythonAI” i el fitxer “visionAI.js” conformen totes les funcions encarregades d’analitzar imatges i de relacionar un codi de barres amb el producte corresponent. És aquí on es troba el model exportat d’etiquetatge personalitzat i l’script amb Python que permet la interacció. Per altra banda, els fitxers “mailFunctions.js”, “mailTemplates.js” i “dbFunctions.js” gestionen i implementen totes les funcions relacionades amb la connexió a Google Firebase, el registre i l’inici de sessió, els assoliments, la generació de codis aleatoris per les recompenses i l’enviament de correus electrònics després de canviar tokens per algun producte. Per últim, el contingut de la carpeta “Public” és temporal i s’utilitza per gestionar les imatges descarregades d’Internet o bé per desar temporalment la imatge que l’usuari envia.

**APP**

Per la part del front-end del sistema hi ha dues parts ben diferenciades: el directori “public” i el directori “src”. Els directoris “Android” i “iOS” inclouen els fitxers que són necessaris per la execució de forma nativa de l’aplicació.

Per una banda, tal i com es mostra a la figura XX, en el directori “public” hi ha tres sub-directoris que inclouen imatges, fonts i les traduccions dels textos a les tres llengües suportades. La resta són diferents fitxers JSON que inclouen tota la informació que s’ha de mostrar per pantalla a l’aplicació. El fitxer “achievements.json” inclou la informació referent a les petites recompenses per utilitzar l’aplicació, el fitxer de “deixalleries.json” inclou la informació referent a totes les deixalleries de Catalunya. Aquesta informació ha estat extreta de la web oficial de la Agència de Residus de la Generalitat de Catalunya []. També, hi ha un fitxer anomenat “faqs.json” que inclou les preguntes més freqüents sobre reciclatge, i els fitxers “products.json” i “recycle.json” inclouen tots els detalls dels residus i contenidors que es poden trobar dins l’aplicació.

Per l’altra banda, tal i com es mostra a la figura XX2, el directori “src” conté diversos arxius que generen la interfície d’usuari de l’aplicació. Els directoris de “router”, “fonts” i “styles” inclouen els fitxers que fan possible la navegació entre els diferents components de l’aplicació i posen l’estil a la interfície d’usuari. Pel que fa al directori “components”, aquí es troben totes els components de l’aplicació, és a dir, totes les pantalles que hi ha incloses dintre de les diferents pestanyes que té l’aplicació. De fet, aquestes pestanyes són “Inici”, “Cerca”, “Configuració” i “Informació” que es troben dissenyades i implementades al directori de “Views”. Per últim, els fitxers “App.vue” i “Main.vue” generen la façana principal de l’aplicació i permet la interacció entre la resta de fitxers que la conformen.