

**only
bites**

NAG Taldea

Gorka Garcia

Naroa Maiztegui

Aimar Alonso

Aurkibidea

Aurkibidea.....	1
Sarrera.....	2
Eskaera.....	2
Proposamena.....	3
Antolakuntza.....	4
Zerbitzari egitura.....	5
SSL.....	5
Firewall konfigurazioa.....	6
MySQL konfigurazioa.....	8
Backup eta cron konfigurazioa.....	9
Django proiekta Linux-en.....	10
Django Apache.....	10
Diseinua.....	12
Kolore paleta.....	12
Tipografiak.....	13
Orrialdeen diseinua.....	13
Figma diseinua eta Estilo gida.....	16
Web aplikazioa.....	17
Proiektuaren egitura.....	17
Datu basea.....	18
Oinarrizko bistak.....	20
Funtzio dinamikoak (JS + JQUERY + AJAX).....	21
Erabiltzaile kudeaketa.....	22
Saskia.....	24
Balorazio sistema.....	25
Multilanguage.....	26
Administrazio panela.....	27
API.....	29

Sarrera

Eskaera

Proiektuak aplikazio bat garatzea eskatzen du, non ingurune digital batean bezeroen eta produktuen informazioa administratzearrekin lotutako funtzionalitate batzuk integratu eta kudeatuko diren. Helburu nagusia bezeroen, produktuen eta erosketen datuak modu eraginkor eta seguruan erregistratu eta antolatzea ahalbidetuko duen sistema bat implementatzea da. Horretarako, datu-base egituratu bat diseinatzea eta programatzea eskatzen da, informazio hori bertan biltzeko, eta haren osotasuna eta irisgarritasuna bermatzeko. Aplikazioak interfazeak izan behar ditu, erabiltzaileari datuak konsultatzeko eta manipulatzeko erraztasunak emateko (informazioa bilatzea eta iragaztea, adibidez), eta datuak behar bezala sartzea eta eguneratzea bermatzen duten neurriak izan behar ditu.

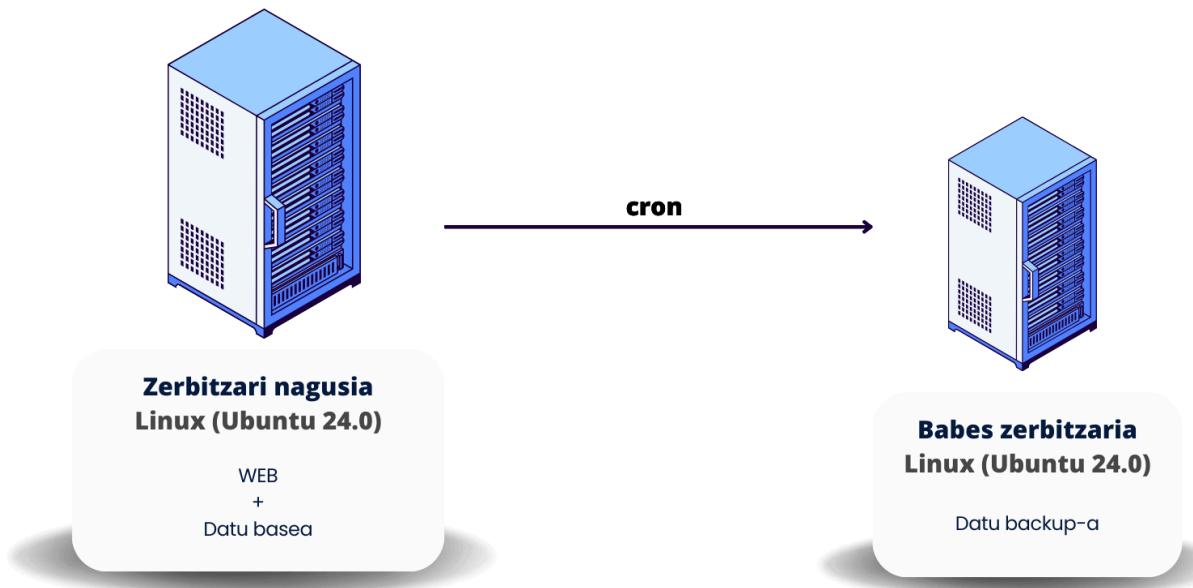
Sistema horren helburua da inventarioak administratzeko eta bezeroak kudeatzeko irtenbide integral bat sortzea, eta, hala, erabiltzaileek informazioaren ikuspegi argi eta antolatua izatea, eragiketen eraginkortasuna hobetzea eta datuak zehaztasunez erabiltzen direla ziurtatzea, erabiltzailearen esperientzia eta sistemaren funtzionaltasuna, oro har, optimizatzeko.

Proposamena

Gure proposamena django framework-a erabiliz, eta beste tresna batuetaz baliatuz webgune bat garatzea da hurrengo feature bereziekin:

- Login with google
- Balorazio sistema bat
- Filtroak
- Subskripzio sistema
- Stock kudeaketa
- Estatistikak

Gure zerbitzarien egiturako proposamena hurrengoa da:

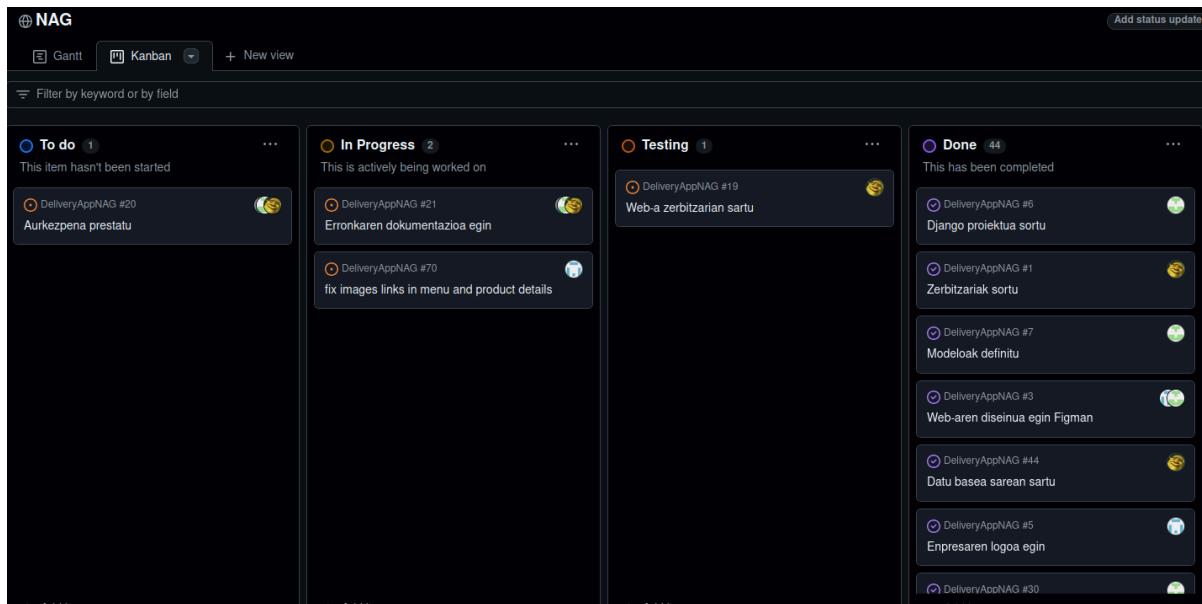


Antolakuntza

Hiru pertsonako talde batean antolatzen gara, eta kide bakoitzari rol jakin bat esleitzen diogu, ardura eta konpromiso indibidualekin. Hori guztia talde-kontratu batean gauzatu zen, gure helburu partekatuak eta bakoitzaren betebeharra zehazten zituena, lankidetza egituratu eta eraginkorra ziurtatuz.

Talde-lanerako, GitHub erabiltzen dugu gure lankidetza-tresna nagusi gisa. Main izeneko adar nagusi bat ezarri genuen, garapen orokorrerako adar develop batekin batera. Gainera, berariazko adarrak implementatzen ditugu, hala nola, feature/... funtzionalitate berriak garatzeko, fix/... akatsak zuzentzeko, eta style/... estiloa doitzena. Egitura horrek aukera eman zigun paraleloan lan egiteko eta aldaketa bakoitzaren kontrol zehatzia izateko.

Zereginak antolatzeko eta haien jarraipena egiteko, GitHub Projects erabiltzen dugu, Kanban metodologia (zereginak denbora errealean bistaratzea eta kudeatzea erraztu ziguna) aplikatuz, eta Gantt, proiektuaren fase bakoitza planifikatzeko eta denborak esleitzeko. Gainera, egunero bost minutuko bilera labur bat egiten dugu jardunaldiaren hasieran, non bakoitzak eguneko aurrerapenak eta zereginak partekatzen zituen, lana doitzena eta zalantzak azkar eta eraginkortasunez argitzeko aukera emanez.



Zerbitzari egitura

Bi makina birtual konfiguratu ditugu, bat zerbitzari nagusia izango da; bertan DB eta webgunea egongo dira. Babes zerbitzari gure datuen backup-a izango du, cron bitartez datuak automatikoki zerbitzari nagusitik babes zerbitzarira pasatzeko.

Hauek izango dira gure zerbitzarien ip-ak:

192.168.73.30 (zerbitzari nagusia)
192.168.73.28 (babes zerbitzaria)

Zerbitzariak ubuntu konfiguratu ditugu eta apache erabili da web orrialdea hostearako, horretarako apache instalazioa pausoak jarraitu ditugu.

SSL

Webgune seguru bat dela adierazteko eta https bitartez sartu ahal izateko ssl segurtasun geruza sartu dugu gure konfigurazioan.

SSL (Secure Sockets Layer) segurtasun-protokolo bat da, **nabigatzailearen eta zerbitzariaren arteko komunikazioa babesteko** erabiltzen dena.

Orain 3 urterako (1095 egun) balioko digun ziurtagiriak sortuko ditugu

Apachen SSL izateko horren modulua instalatu dugu.

Gero virtualhosteen konfigurazioaren oinarrizko zatia egin dugu

onlybites.eus.conf-en lotura sinbolikoa egin dugu eta konfigurazioa gaitzen dugu

Ondoren ezgaitu egiten dugu aurrezarrita datorren gunea, honela localhostean agertzen zen defektuzko orria ez da agertuko guk nahi dugun orria baizik:

Host listaren barruan hurrengo localhost-a duen IP eta webgunearen domeinua jarri dugu

localhost IP

dudan domeinuaren izena jarrita

Honela geratu da:

127.0.0.1 localhost

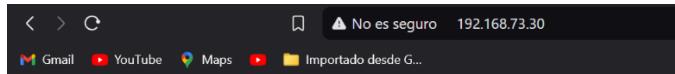
127.0.1.1 wag-Standard-PC-Q35-ICH9-2009

127.0.0.1 onlybites.eus

sartzen dut baina nahi

Lokalean berdina egiten dugu baina 127.0.0.1 jarri beharrean makina birtualaren IP-a jarri dugu

Orain makina birtualaren webgunea ikusteko



OnlyBites

Nabegatzailean ohar bat jarriko dizu, baietz ematen badiozu zure webgunea aterako zaizu

Firewall konfigurazioa

Zerbitzaria babesteko, beharrezkoak diren portuak bakarrik izango dute firewallaren baimena, hala nola 80(HTTP), 443(HTTPS) eta 81(Test HTTPS) portuak.

Hasteko firewall-a baimentzeko dugu

Gero nahi ditugun portuak(80,81,443) firewallaren baimena ematen diogu TCPrekin

Datu Basera kanpotik konektatzeko bakarrik baimendutako IPak sartuko dira

```
wag@wag-Standard-PC-Q35-ICH9-2009:~$ sudo ufw status
Estado: activo

Hasta          Acción      Desde
-----        -----
80/tcp          ALLOW      Anywhere
443           ALLOW      Anywhere
443/tcp         ALLOW      Anywhere
3306           ALLOW      192.168.73.30
3306           ALLOW      192.168.73.26
3306           ALLOW      192.168.73.9
3306           ALLOW      192.168.73.8
80/tcp (v6)    ALLOW      Anywhere (v6)
443 (v6)       ALLOW      Anywhere (v6)
443/tcp (v6)  ALLOW      Anywhere (v6)

wag@wag-Standard-PC-Q35-ICH9-2009:~$
```

MySQL konfigurazioa

MySQL Workbench erabili datu basea errazago kudeatzeko

Gero mysql-eko erabiltzaile berri bat sortu dugu privilege guztiekin

Mysql-en erabiltzaile berri bat sortu dugu eta horri Privilege guztiak ematen diogu

Gero erabiltzaile horri Privilege guztiak ematen diogu

Honela Datu Basera konektatzeko:

Hostname:localhost

Port:3306

Username:admin

Password:Admin123

Kanpotik konektatzeko

Mysql konfiguratu dugu datu basera kanpotik konektatzeko

Kanpotik konektatzeko hostname aldatzen dugu, makina birtualaren IP-a jarriz

Hostname:192.168.73.30

Backup eta cron konfigurazioa

Backup-a egingo duen sh-an sartzen

sudo nano Escritorio/backup.sh

Barruan honela geratu beharko litzateke:

```
GNU nano 7.2                                         Escritorio/backup.sh
#!/bin/bash
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# Variables
$DB_NAME="nag"
BACKUP_DIR="/home/wag/Escritorio/backup"
BACKUP_FILE="$BACKUP_DIR/backup_$(date +%F_%H-%M).sql"
LOG_FILE="/home/wag/backup.log"

# Create backup
echo "Inicio del backup: $(date)" >> $LOG_FILE
mysqldump $DB_NAME > $BACKUP_FILE 2>> $LOG_FILE

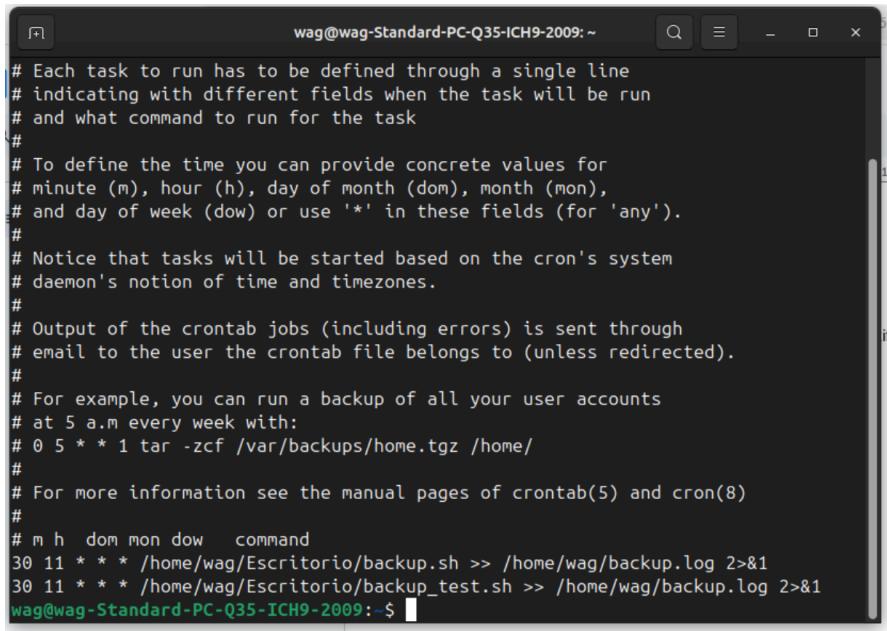
# Verificar if the backup was correctly created
if [ $? -eq 0 ]; then
    echo "Backup creado correctamente: $BACKUP_FILE" >> $LOG_FILE
else
    echo "Error al crear el backup" >> $LOG_FILE
    exit 1
fi

# Transfer the backup to another virtual machine
echo "Transfiriendo el backup a la máquina remota..." >> $LOG_FILE
scp $BACKUP_FILE wag@192.168.73.28:/home/wag/backup/ 2>> $LOG_FILE

# Verify if the transfer was done successfully
if [ $? -eq 0 ]; then
    echo "Transferencia completa." >> $LOG_FILE
else
    echo "Error al transferir el backup" >> $LOG_FILE
    exit 1
fi
```

Test datu basearen backup egiteko scripta sortu dugu

Gero crontabean bi backup-ak egiten dituzten scripten kokapenak sartzen ditugu eta scripta zer ordutan eta frekuentzia jartzen dugu:



```
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
30 11 * * * /home/wag/Escritorio/backup.sh >> /home/wag/backup.log 2>&1
30 11 * * * /home/wag/Escritorio/backup_test.sh >> /home/wag/backup.log 2>&1
wag@wag-Standard-PC-Q35-ICH9-2009: ~
```

Kasu honetan 2 script-ak egunero exekutatuko dira 11:30an eta log batean gordeko dira aldaketa guztiak errore bat egongo balego jakiteko arrazoia

Django projektua Linux-en

Errepositorioa klonatzen dugu

Ondoren entorno virtuala instalatu dugu komando

Gero terminalean sartzen gara eta errepositorioaren carpetara sartzen gara
Entorno birtuala sortu eta martxan jartzen dugu:

Proiektuaren dependentziak instalatu ditugu

Gero datu basea migratzen dugu:

Gero, sare berdinean dauden ordenagailuak gure webgunea ikusteko virtualhosta sortu
dugu

Gero gure webgunea bere funtzio guztiekin ikusteko 443

Django Apache

Gero, Django Webgunea Apachera mugitu eta django apacherekin funtzionatzeko wsgi
liburutegia instalatu dugu

Modulu honi esker, Apache **Python** aplikazioak zerbitza ditzake (bereziki web aplikazioak)
WSGI protokoloaren bidez (Web Server Gateway Interface). Protokolo hori web
zerbitzariak Python aplikazioekin konektatzeko interfaze estandarra da.

Gero gure domeinuaren apache konfigurazioan hurrengo aldaketak egiten ditugu django
proiektua zerbitzatzeko, honekin batera testeatzeko virtualhost bat gehituko dugu.
Horretarako django proiektua klonatuko dugu, baina virtualhost berri honetako django
proiektua develop branchein egongo da eta hasieran sortu genuen virtualhost-arena
main branchein egongo da.

Hiru virtualhost
desberdinietan:

dauude portu

- **80 portua:**Gure webgunearen izena jarri dugu servername bezala eta http bezala sartzen badugu url-a https ra bihurtuko du, honekin sare berdinean dauden ordenagailua guztiak konektatu ahalko dira, nahiz eta googlekin saioa hasi ezin izan

```
# Redirección de HTTP a HTTPS
<VirtualHost *:80>
    ServerName onlybites.eus
    Redirect permanent / https://192.168.73.30/
    DocumentRoot /var/www/DeliveryAppNAG
</VirtualHost>
```

- **443 portua:**Main webgunera bideratuko digun Https konfigurazioa da.Honetan django proiektuaren lokalizazioa sartutako dago, virtualhostak hainbat karpetetera sartzeko baimenak daude sartuta baita static karpetarako, non gure webgunearen estiloa eta hainbat funtzio daude gordeta.
- **81 portua:**443 portuaren konfigurazioaren kopia bat da, aldaketa bakarra djangoen lokalizazioa izango da, non develop branchean dagoen karpetaren lokalizazioa erabiliko dugu

```
<VirtualHost *:443>
    ServerName onlybites.eus
    ServerAdmin admin@onlybites.eus

    DocumentRoot /var/www/DeliveryAppNAG
    # Configuración de WSGI
    WSGIDaemonProcess onlybites python-home=/var/www/DeliveryAppNAG/Nag python-path=/var/www/DeliveryAppNAG/onlybites/onlybites
    WSGIProcessGroup onlybites
    WSGIScriptAlias / /var/www/DeliveryAppNAG/onlybites/onlybites/wsgi.py
    # Pasar encabezados de autorización a la aplicación WSGI
    WSGIPassAuthorization On
    # Seguridad y permisos
    <Directory /var/www/DeliveryAppNAG>
        Require all granted
    </Directory>

    Alias /static /var/www/DeliveryAppNAG/onlybites/onlybites/staticfiles
    <Directory /var/www/DeliveryAppNAG/onlybites/onlybites/staticfiles>
        Options FollowSymLinks
        Allow from all
        Require all granted
    </Directory>

    Alias /media /var/www/DeliveryAppNAG/onlybites/onlybites/media
    <Directory /var/www/DeliveryAppNAG/onlybites/onlybites/media>
        Options FollowSymLinks
        Allow from all
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on

    SSLCertificateFile      /etc/apache2/ssl/server.crt
    SSLCertificateKeyFile  /etc/apache2/ssl/server.key
```

Aipatutako virtualhosteen portuak entzuteko ports.conf honela geratuko da:

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
<IfModule ssl_module>
    Listen 443
    Listen 81
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

Diseinua

Janari delivery webgune honetarako diseinu garbi, lasai, orientatibo eta modernoa planteatu da.

Kolore paleta

Kolore paletari dagokionez, bi multzotan sailka ditzakegu hautatutako koloreak emandako erabileraren arabera.

Kolore bizi eta deigarriak (laranja eta berdea) webguneko botoi eta testu garrantzitsuetarako.

Laranjak transmititzen duen bizitasuna eta inputsibitatea ederki orekatzen da berdearen lasaitasunarekin.

Kolore argi eta lasaiak webgunearen atzeko planorako eta atalen arteko bereizketa egiteko tresna gisa erabili dira. Hauei esker bezeroaren arreta testu eta webgunea osatzen duten gainontzeko elementuetara bideratuko da.



Kolore beltza erabili da testu batzuen irakurgarritasuna optimizatzeko, eta kolore zuria botoien testuetarako.

Kolore ezberdinak aplikatzeak oreka bisual ona mantentzea ahalbidetzen digu, erabiltzaileari kolore karga ekidinez.

Tipografiak

Hiru tipografia desberdin erabili dira, bi nagusi eta bigarren mailako bat, xehetasun txikietan bereziki aplikatu dena. Tipografien pisuekin jolastu egin da weguneko testuei garrantzia kendu eta emateko.

- **Hagrid text:** Nortasun handiko tipografia modernoa da, gorputz zabala eta trazuen artean kontraste gutxikoa. Webguneko atalen sarrera testu eta izenburuetan aplikatu da. Hagrid Text Zetafonts-ek eskainitako lizentzia komertziala du.
- **DM sans:** Tipografia modernoa, garbia eta oso irakurgarria. Gainerako tipografiarekin kontrastatzen du bere sinpletasunean. Testu luzeetan erabili izan da bereziki. Open Font lizentziapean dago baimenduta.
- **Playfair Display:** Itxura sofistikatuagoa duen tipografia da. Trazuen arteko kontraste handiak izaera klasiko eta modernoa ematen dio. Produktuen informazio horriean erabili da bereziki. Open Font lizentziapean dago baimenduta.

Orrialdeen diseinua

Diseinuarekin hasi aurretik oinarrizko orrialdeak definitzen hasi ginenean: **home, menu, product details, perfil, saskia, login eta erregistratzeko** orriak.

Lehenik eta behin, gris-eskalan diseinatu genituen orriak, edukia nola egituratuko den ikusteko.



Hasierako orriaren diseinuaren oinarria gris eskalan.

Orri guztien oinarria diseinatu ondoren, diseinuari izaera ematen hasi ginen, ezarritako estilo-gida jarraituz. Webguneko elementu guztiak gehituz: botoiak, logoa, irudiak, testua , etb.



Erraza, goxoa eta zuzenean zure mahaira.
Janari prestatu aberats eta osasunberria.

Bezeroen gustukoenaak

Aste honetan gure bezeroei gehien gustatuko bite-ak probatzen ahal dira:

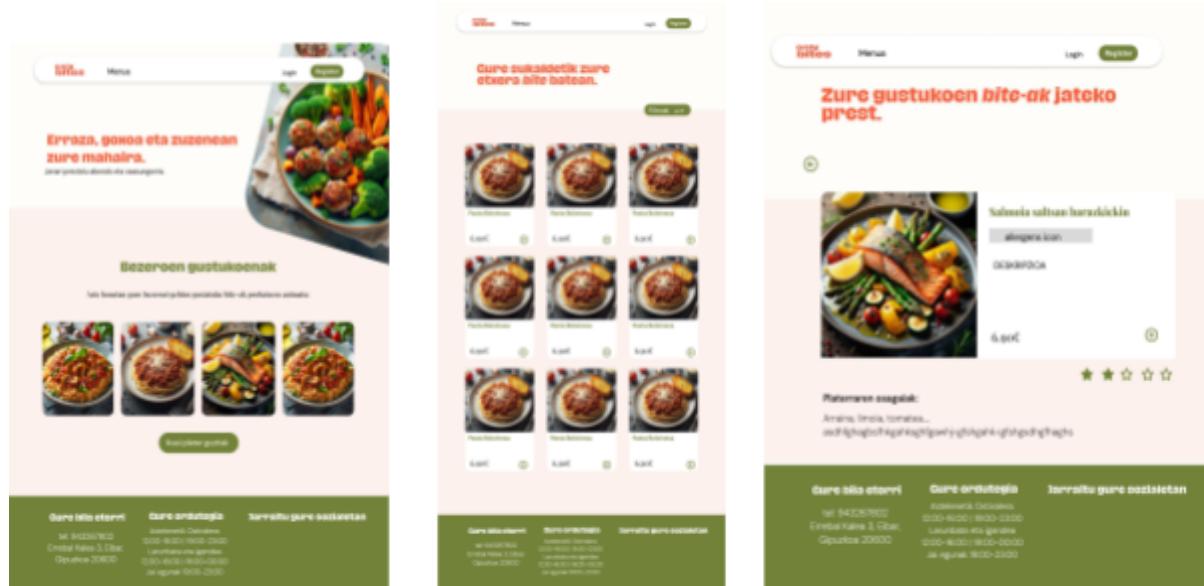





[Ikusi plater guztiak](#)

Gure bila etorri tel: 943267802 Errebal Kalea 3, Elbar, Gipuzkoa 20600	Gure ordutegia Astelenetik Ostrailera: 12:00-16:00 19:00-23:00 Larunbata etan igandeara 12:00-16:00 19:00-00:00 Jai egunak 19:00-23:00	Jarraitu gure sozialetan
--	--	---------------------------------

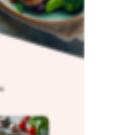
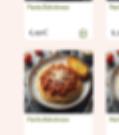
Hasierako orriaren diseinu definitiboa.



Erraza, goxoa eta zuzenean zure mahaira.
Janari prestatu aberats eta osasunberria.

Bezeroen gustukoenaak

Aste honetan gure bezeroei gehien gustatuko bite-ak probatzen ahal dira:



[Ikusi plater guztiak](#)

Zure sukaldetik zure etxera bite batzuen.

Zure gustukoenaen bite-ak jateko prest.



Salmoia sultza harruskidetik
[alergoia leinu](#)

[Detaldeak](#)

6,99€

Potenzialen espezieak:
Aneto, lima, tomate...
osztalj meg a legjobbat a legjobbat!

Gure bila etorri tel: 943267802 Errebal Kalea 3, Elbar, Gipuzkoa 20600 Jai eguna 19:00-23:00	Gure ordutegia Astelenetik Ostrailera: 12:00-16:00 19:00-23:00 Larunbata etan igandeara 12:00-16:00 19:00-00:00 Jai egunak 19:00-23:00	Jarraitu gure sozialetan
---	--	---------------------------------

The screenshots show a user profile creation page, a menu selection page, and a payment confirmation page.

- Kaixo, (izena)**: A form for creating a user profile. It includes fields for Name (Izena), Surname (Aizkaina), Phone number (Telefono handilean), Date of birth (Jaiotze data), and Password (Pasitzua). Below the form is a section for Delivery addresses (Entzute helbideak) with two entries: "20000 Gipuzkoa, Elorri, Zubiarreta 6, 01000" and "20002 Gipuzkoa, Elorri, Zubiarreta 6, 01000". Buttons for "Gorde" (Add) and "Gorde" (Save) are present.
- (Izena)-ren saskia**: A product selection page showing two items:
 - Pezzi bellesse**: Price 6,90€, with a "Buy" button.
 - Natxitua salteko herrekindeka**: Price 6,90€, with a "Buy" button.
 Below the products is a section for Delivery address (Entzute helbideak) with the entry "20000 Gipuzkoa, Elorri, Zubiarreta 6, 01000". A "Gorde" (Add) button is shown.
- Sorktu kontua**: A payment confirmation page. It shows the same delivery information and a "Gorde" (Save) button.

Oinarrizko orrialdeen diseinua.

Ikusi daitekenez egitura homogeneoa lortu da webgunean zehar, elementu guztiei antzeko itxura emenez.

Webguneko orri nagusi guztien hasieran (home, menu eta product details) nortasun propioa ematen dion esaldiak ditugu. Aldiz, gainerako orrialdeen goiburuak (saskia, perfil eta erregistratzeko orria) erabiltzaileari zuzenduta daude, webgunearen eta erabiltzailearen arteko hurbiltasun-harremana lortzeko helburuarekin.

Koloreen erabilerari dagokionez tresna nagusi bezala erabili da webguneko atalen eta orrialdeen arteko elementuen arteko desberdintasuna lortzeko. Orrialde nagusiei bi koloretako atzealdea aplikatu zaie sarreretako esaldien eta produktuen artean bereizketa egiteko. Izan ere perfil eta saski orrialdetan lerro mehe beltza arduratzen da atalen bereizketa egiteaz.

Kolore berriko botoiak implementatu ditugu, botoi bakoitzaren funtzioa bere kolorearekin bat izan dadin.

- Berdea:** neutroa, gehien bat besteista batera eramatzen gaitu, pop up formularioak azaltzeko baita ere.
- Laranja:** aldaketa bat adierazteko. Erosketa helbideko datuak editatzeko eta ordainketa egiteko erabili izan da.
- Gorria:** alerta gisa. Helbidea zein perfilak ezabatzeko botoietan.

Figma diseinua eta Estilo gida

Hurrengo esteketan, web-aren estilo-gida eta diseinuak (pantaila mugikorra eta ordenagailukoa) aurkitu daitezke.

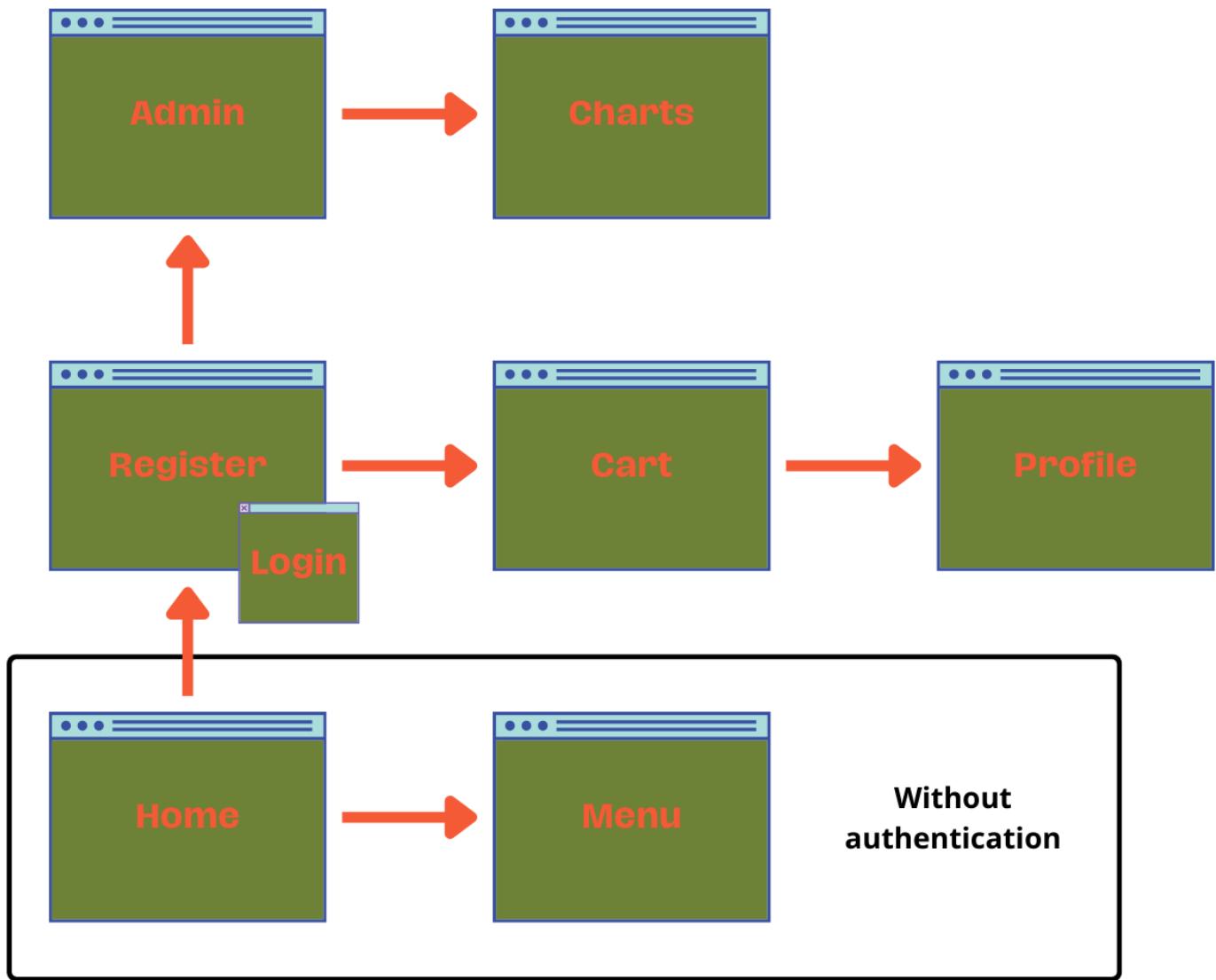
Webgune diseinua [hemen](#).

Estilo gida [hemen](#).

Web aplikazioa

Proiektuaren egitura

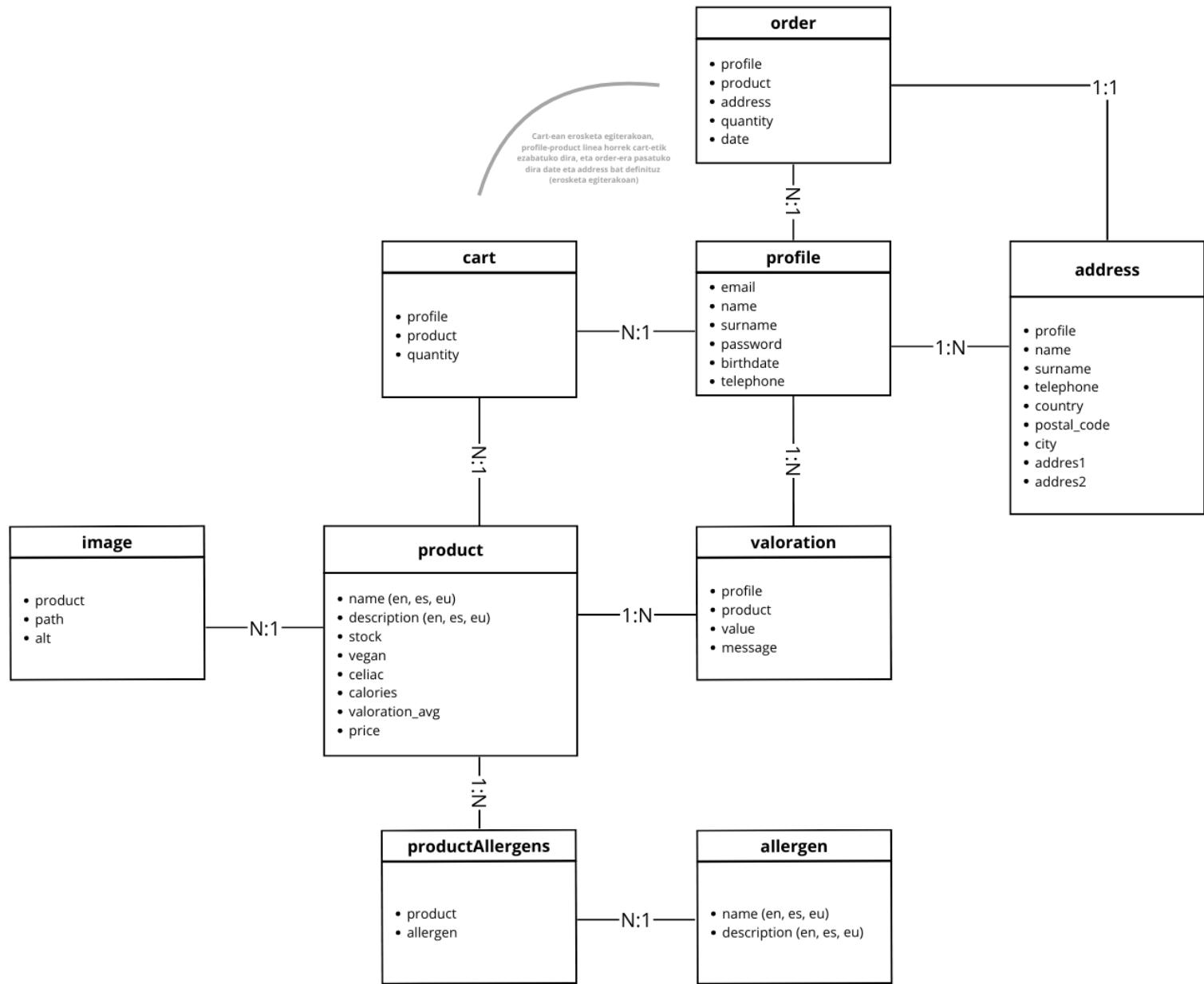
Proiektua garatzen hasi aurretik, honentzako egitura egokiena zein zen pentsatzen aritu ginen. Lehenik eta behin, view orokorrak eta haien arteko ekintzak zein izango ziren pentsatu genuen, eta horietako bakoitzean zer erakutsiko genuen:



Web aplikazioaren barne egituraren diseinua.

Datu basea

Gero, proiektuaren modeloetarako egitura bat planteatzen hasi ginen, zer eredu definitu, zer eremu, ereduen arteko erlazioak, eta abar.



Datu basearen egituraren diseinua.

Zenbait erlaziori buruzko oharrak:

- **Cart - Order:** Erabiltzaile batek saskiari gehitzen dizkion produktuak Cart taulan gordeko dira, baina erabiltzaileak ordainketa egiten duenean, Cart-eko datuak

Order taulara pasatuko dira, erabiltzaileak aurredefinitutako helbide bat eta momentuko data eskuratz, eta Cart taulatik ezabatuko dira.

- **Product - Allergen:** Produktu batek alergeno bat baino gehiago izan ditzakeenez eta, aldi berean, alergeno bat produktu bat baino gehiagorena izan daitekeenez, ProductAllergen taula sortzen da produktuak eta alergenoak erlazionatzeko.
- **Product - Image:** Irudi bat hainbat eremu dituenez, egitura garbiagoa geratzen da, eremu horiek beste Image taula batean definituz gero, gainera, horrek projektua eskalagarriago egitea ahalbidetzen du, produktu bakoitzeko irudi bat baino gehiago izateko aukera eskaintzen duelako eta produktuetarako irudi galeria bat implementatu ahal delako.

Oinarrizko bistak

Proiektuaren oinarrizko bistak garatzeko eta, horrela, talde-garapenari jarraitzea errazagoa den puntu bat izateko, djangok eskaintzen dituen oinarrizko egitura eta tresnak erabiltzen ditugu:

- Proiektu bat sortzean django ematen duen models.py fitxategia erabiliz, gure taulak python-eko objektu gisa definitzen ditugu, CRUD eragiketak askoz modu errazagoan egin ahal izateko eta prozesu osoa definitu behar gabe datu-basean aldaketak egiteko (connector, cursor, query, ...).
- Fitxategi estatikoen eta django templatuen egitura erabiliz, frontend-aren garapena isolatu dezakegu, eta, gainera, templatuen barruan logika aplikatzeko aukera ematen duten tag mota desberdinak erabiltzeko aukera ematen du.
- urls.py eta views.py fitxategiekin, ikuspegien eta txantiloien arteko oinarrizko ekintzak defini ditzakegu.
- forms.py fitxategia erabiltzen dugu, urls.py eta views.py-rekin batera, erabiltzailearentzat instantzia berriak sortzeko aurrez definitutako formularioak sortzeko.
- Django eskaintzen duen administrazio panela erabiliz, ia lan guztia eginda daukagu tresna bat garatzeko orduan, webgunearen administratzaileari erakusten eta kudeatzen den eduki guztia kudeatu ahal izateko.

Laburbilduz, framework honek eskaintzen dituen tresna guztiak erabiltzen ditugu MVC (Model-View-Controller) arkitektura duen web aplikazio bat garatzeko, garatzeko eta mantentzeko errazagoa dena, atal bakotza gainerako kodeetatik askoz isolatuago dagoelako eta beharrezkoa den scope-a identifikatzea askoz errazagoa delako.

Funtzio dinamikoak (JS + JQUERY + AJAX)

Proiektu horretan JavaScript, AJAX eta jQuery erabili ditugu gure web aplikazioan erabiltzailearen interakzioa eta esperientzia hobetzeko, informazioa egunearatzea ahalbidetuz orrialde osoa kargatu beharrik gabe. Batez ere, iragazkien implementazioan, popup-en bistaratzean eta datuen erabilera dinamikoan aplikatu ditugu.

Produktuen iragazkietarako, hainbat aukera definitzen ditugu, hala nola vegan, celiac, maxCalories eta allergies selected_filters objektu batean, erabiltzailearen hautaketen arabera egunearatzen dena. Erabiltzaileak iragazki batekin elkarreragiten duenean, updateProductList funtzioari deitzen zaio. Funtzio horrek AJAX erabiltzen du iragazkietako datuak zerbitzarira bidaltzeko eta produktuen zerrenda egunearatua lortzeko, orria kargatu gabe.

Era berean, AJAX eta jQuery erabiltzen ditugu popupak kudeatzeko, hala nola login formularioa eta helbidearena. Login botoian edo helbidea gehitzeko botoian klik egitean, AJAX funtzi bat exekutatzen da, zerbitzaritik popup-aren edukia kargatu eta HTMLan txertatzen duena. Leihoriek erabiltzaileari aukera ematen diote logina edo helbide-aldeketa bezalako ekintzak osatzeko, aldeketak orria kargatu gabe gordez. Datu horiek gordetzeko, saveAddress funtzioak, adibidez, formularioa serializatu eta AJAX bidez bidaltzen du, interfazea datu berriekin egunearatuz.

Gainera, saskian, CartQuantity, incrCartQuantity eta deleteCart bezalako funtziok produktuen kantitatea dinamikoki kudeatzen dute, AJAX erabiliz saskia denbora errealean egunearatzeko. Kodearen egitura gertaera eta funtzi espezifikoetan oinarritzen da, eta erabiltzailearen interakzio bakotza interfazeko erantzun azkar eta eraginkor bihurtzen dela ziurtatzen du.

Erabiltzaile kudeaketa

Erabiltzaileen kontu gisa erabiltzeko eredu propio bat garatu genuen hasiera batean. Horretarako, erabiltzaileak kudeatzeko dagozkion CRUD aukerak egin genituen:

- Erregistroa: Erabiltzaile-instantzia bat sortu, autentifikatu gabeko erabiltzaile baten eskura dagoen formulario baten bidez.

Erregistro formularioa.

- Login: Sartutako kredentzialak datu-basean erregistratutako erabiltzaile-instantziekin alderatzea, eta bat etorri gero, djangok eskaintzen dituen tresnak erabiltzailearen nabigatzailaren saioko cookieak definitzeko eta erabiltzailea autentifikatzeko.

Login formularioa.

- Kontua ezabatu: Aukera bat sortu erabiltzaileak bere kontua ezaba dezan, gure datu-basean bere datuak mantentzen jarraitu nahi ez badu.

Ondoren, erabiltzailea login with google erabiliz logeatzeko aukera implementatzen hasi gin, honentzat askoz erosoa eta seguruagoa dena, ez baititu hainbat pasahitz gogoratu eta kudeatu behar.

Implementazio hau nahiko erraza da, egin behar den gauza bakarra erabiltzailea bere google kontuen aukeraketara bideratzeko URLak erabiltzea baita, eta erabiltzailearen saioa sortzeko, saioa hasteko eta ixteko bistak definitzea.

Login with google egitean, django lehenetsitako erabiltzaile-taulan gordetzen dira datuak. Taula hori erabat independentea da gure proiekutuko taularekiko, zeinarekin erabiltzaileen kudeaketa egiten genuen.

Horregatik, proiekutan hobekuntza batzuk egitea erabaki genuen, guk garatutako eredu django eredu lehenetsia izan zedin. Hori egiteko, aldaketa hauek egin behar izan genituen:

- Manager eredu bat sortu, django BaseUserManager eredu zabaltzen duena, erabiltzaileak eta supererabiltzaileak nola sortu behar diren definitzeko.
- Gure erabiltzaile-modeloa (Profile) django (AbstractBaseUser eta PermissionsMixin) bidez aurrez definitutako erabiltzaile-klaseak heredatzea.
- Aldatu settings.py lehenetsitako erabiltzaile-modeloak zein izan behar duen definitzeko, gehitu bi erroregistrotarako middlewareak, definitu birbideratze-bistak login eta logout egitean, definitu Googleren apiaren key-ak, ...

Hori guztia lortu ondoren, erabiltzailearen kudeaketa egin dezakegu gure beharretara egokitutako eredu batekin, garatzaileari zein erabiltzaileari lana errazten dieten django eta google tresna aurredefinituak erabiliz.

Saskia

Saskia da erabiltzaileak erosi nahi dituen produktuak gordetzen lekua, eta aurrerago ordainketa egin eta produktuen erosketa amaitzen aukera eskainiko duen bista.

Produktu bakotzak, bai menuaren ikuspegian, bai produktu baten ikuspegi xehatuan, erabiltzailearen saskira gehitzeko aukera du. Aukera hau plus batekin bistaratzan da.



Ensalada mixta

100 cal

6.9€



Ensalada mixta

100 cal

Ensalada mixta

6.9€

Vegano ★★★★☆



Nahi ditugun produktuak gehitu ondoren, saskitik gure erosketari buruzko informazioa ikusi ahal izango dugu, baita gure saskia editatzeko hainbat aukera ere, hala nola produktu kopurua handitu eta murriztea, edo produktu bat gure saskitik ateratzea.

Behin erabiltzailea erosketaz ziur dagoenean, erosketa egin dezake, bere helbideetako bat aukeratuz. Erosketa egitean, Carten instantziak Orderrera pasatuko dira.



Carrito

Productos:



1

7,0€



Pagar

Balorazio sistema

Produktu bakoitzaren balorazioak, erabiltzaileek produktu jakin horri buruz duten iritzia uzteko aukera bat da, eta aldi berean beste erabiltzaileek iritzi horiek kontuan hartzeko produktu bat edo bestea aukeratzerakoan.

Produktu bakoitzaren xehetasunen ikuspegian, beheko aldean, foro moduko bat dago, eta han ikus daitezke produktuari buruzko iritzi guztiak, haien mezua eta balorazioa (bostetik).

Produktuak valoration average eremua du, eta hor erabiltzaileen balorazioen batez bestekoa agertzen da.

signals.py (trigger moduko bat) fitxategiaren bidez, baloration instantzia berri bat erregistratzen den bakoitzean, produktuaren baloration averagea kalkulatzen eta eguneratzen duen funtzio bat exekutatzen da.



Balorazioak:
Gorka: ★ ★ ★ ★ (1/5) - ghjkgjhjk

Baloratu

x

Entsalada

Balorazioa ★ ★ ★ ★

Balorazioa

Mezua:

Oso ona!

Bidali balorazioa

Multilanguage

Kontuan hartuta gure zerbitzua nazioarteko dela, baina salmentarik handienak estatu mailakoak direla, zehazki Euskadikoak, bezeroek feature extra gisa eskatu zuten hizkuntzak aldatzeko aukera gehitzea, gutxienez Ingelesa, Gaztelania eta Euskara.

Garapen hori gauzatzeko, lana bi zati nagusitan banatu genuen:

- Webguneko testu estatikoaren itzulprena.
- Ereduetik kargatzen den testu dinamikoaren itzulprena.

Testu estatikoa itzultzeko, Djangok ematen dituen baliabideak erabiltzen ditugu, eta, horri esker, hainbat .po fitxategi sor ditzakezu kanpoan, hizkuntza bakoitzerako itzulpenak definitzeko. Hori lortzeko prozesua nahiko erraza da:

1. Konfiguratu hizkuntzak hemen: settings.py.
2. Sortu itzulpen-fitxategiak makemessages komandoa erabiliz.
3. Fitxategien itzulpenak egin.
4. Itzulpenak compilemessages komandoarekin aplikatu.

Ereduen testua itzultzeko zatia zailagoa da, aurrerago eskatu zen extra bat izan zela kontuan hartuta, eta, beraz, datu basea ez zegoen funtzio horretarako egituratuta.

Horregatik, eraginkorrena ez izan arren, baina bai praktikoena, hizkuntza bakoitzerako eremuak sortzea erabaki genuen, beharrezkoak ziren eremuetan.

Itzuli behar genituen eremuak azterketa egin ondoren, eredu hauetan aldaketak aplikatzea beharrezkoa zela definitu genuen:

- **Product:** name eta description
- **Allergen:** name eta description

Sortutako eremu horiekin, templatetatik, saioaren hizkuntza ikus daiteke, eta, aldagai horren arabera, eremu bat edo bestea aukeratzen da.

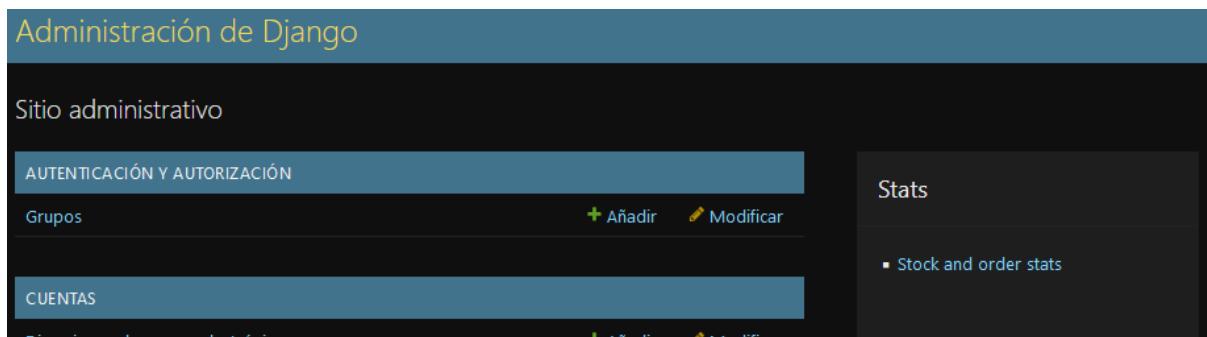
Administrazio panela

Estatistikak webguneko administratzailak enpresaren emaitzetatik ondorioak modu simpleago eta bisualagoan ater ahal izateko bideratuta daude.

Garapen hau webgunearren administraziora bideratuta dagoenez, django eskaintzen duen admin panelean sartzea erabaki genuen.

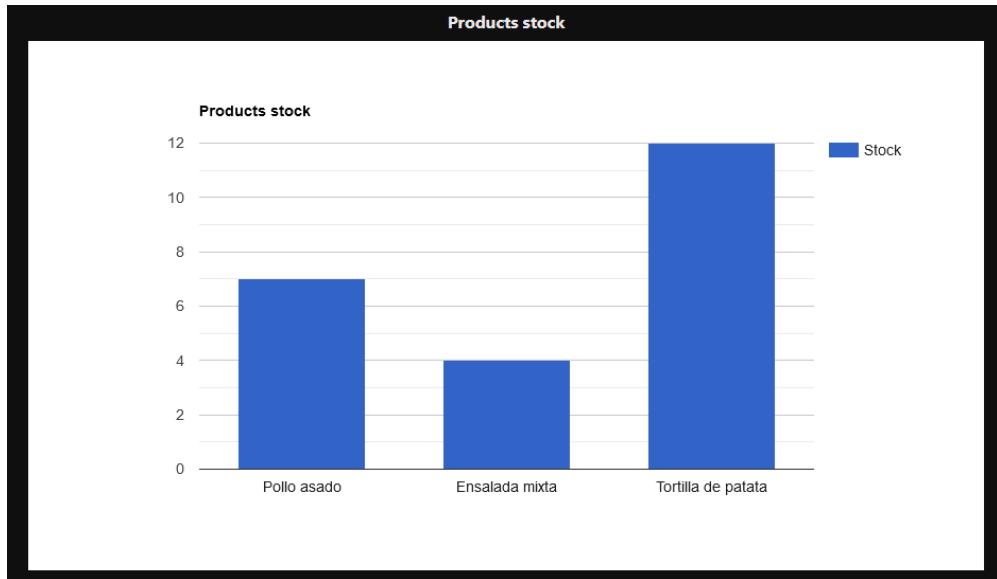
Honen arazoa da django admin aplikazioa ingurune birtualean instalatzen den django paketean dagoela, beraz, hor egiten diren aldaketak, ezin dira errepositoriora igo, edo behintzat, ez da zuzena.

Hori konpontzeko, admin-erako index template propio bat sortzen dugu, eta index template lehenetsia konfiguratzen dugu settings.py konfigurazio-fitxategian. Horrela, admin-aren index ikuspegia nahierara editatu ahal izango dugu.

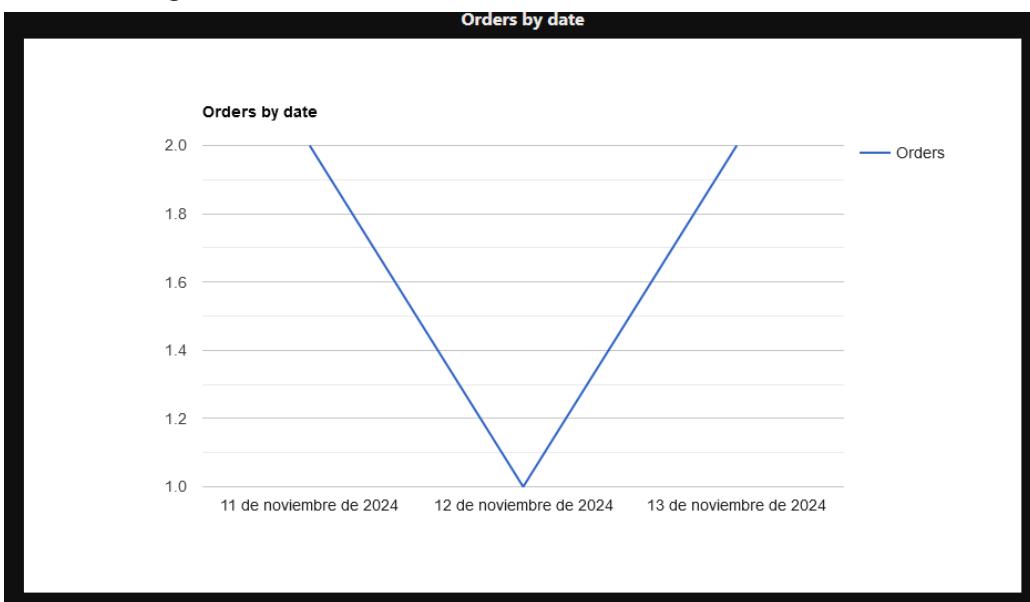


Index-etik, url baterako esteka zuzena sartu genuen, non webgunearren kudeaketari buruzko bi grafiko sartu genituen. URL honetara erabiltzailea supererabiltailea denean bakarrik sar daiteke. Grafiko batean erakusteko aukera interesgarrienak hauek direla erabaki genuen:

- Produkto bakoitzaren stockak jasotzen dituen grafiko bat.



- Salmenten grafiko bat, dataren arabera.



API

Djangoren rest api-a produktuak kudeatzeko erabili dugu.
Bakarrik administratzaileak erabili ahalko dute.

Bi endpoint daude: Bat datu basea orokorrean kudeatzeko eta bestea produktu espezifiko bat kudeatzeko non produktuaren id-a sartzen dugu

HTTP metodo nagusienak erabili ditugu:

- GET
- POST
- DELETE
- PUT

Endpoint generalean erabili ditugun HTTP metodoak:

- GET: Produktu guztien lista bat ikusteko
- POST: Produktu berri bat sartzeko

Endpoint espezifikoan erabili ditugun HTTP metodoak:

- GET:Produktu espezifiko hori ikusteko
- PUT:Produktu hori aldatzeko
- DELETE:Produktu hori kentzeko