



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

ZIENTZIA
ETA TEKNOLOGIA
FAKULTATEA
FACULTAD
DE CIENCIA
Y TECNOLOGÍA



Gradu Amaierako Lana
Ingeniaritza Elektronikoko Gradua

Portaera sozialaren eragina epidemien hedapenean

Simulazio epidemiologikoak burutzeko programaren
diseinu eta inplementazioa

Egilea:
Unai Otamendi Loibide
Zuzendaria:
Mikel Peñagarikano Badiola

Gaien Aurkibidea

Sarrera	1
1 Oinarri teorikoak	3
1.1 Transmisio metodo nagusiak	3
1.1.1 <i>Flügge</i> tanten bidezko transmisioa	4
1.1.2 Aireko partikulen bidezko transmisioa	4
1.1.3 Kontaktu zuzeneko transmisioa	5
1.1.4 Zeharkako kontaktu bidezko transmisioa	5
1.2 Patogeno dentsitate eta esposizioa	5
1.3 Kutsatze probabilitatea	7
2 Programaren eraikuntza	11
2.1 Klase generikoak	11
2.1.1 <i>Person</i> klasea	12
2.1.2 <i>Peoplelist</i> klasea	14
2.1.3 <i>Place</i> klasea	15
2.1.4 <i>Illness</i> klasea	16
2.2 Klase espezifikoak	18
2.2.1 <i>Town</i> klasea	19
2.2.2 <i>Conduct</i> klasea	22
2.2.3 <i>Population</i> klasea	24
2.2.4 <i>Data</i> klasea	28
3 Epidemiaren hedapenaren simulazioa eta emaitzen analisisa	29
3.1 Pandemiaren eboluzioa: Ekaina-Abendua	29
3.2 Aztarnatze metodo eta hedapen abiadura	33
3.2.1 1. Metodoa: Aztarnatze metodo estandarra	34
3.2.2 2. Metodoa: Gertuko kontaktuen zabaltzea	35
3.2.3 3. Metodoa: Ausazko testak	37
3.2.4 Hedapen erritmoa	38
3.2.5 Emaitzen bariantza	40
3.3 Segurtasun metodo pertsonalen eragina	41
3.3.1 Distantziamendu soziala	41
3.3.2 Maskaren erabilera	42
4 Ondorioak	45
Bibliografia	49
Eranskinak	51

Sarrera

2019 urtearen amaieran lehen COVID-19 kasuak detektatu zirenetik, gaixotasun kasuak era exponentzian hazi ziren [1, 2]. Urte baten buruan, 2020 urte amaierarako, 80 milioi kasu eta 1.78 milioi hildako eragin zituen pandemiak mundu osoan zehar, kasu aktiboak 20 milioi baino gehiago izanik [3].

Pandemiaren hedapen abiadura moteltzeko neurri ez-farmazeutikoak funtsezkoak izan dira, distantziamendu soziala edo konfinamendu neurriak esaterako [4]. Hala ere, neurri hauek ezartzeko orduan ez da adostasunik izan, ondorioz, ezarritako neurri asko herrialdearen arabera ezberdinak izan dira [5]. Espainiako estatuan ere, autonomia erkidego bakoitzak irizpide ezberdinak izan ditu mugikortasun eta ordutegi neurriak zehazteko.

Irizpideen arteko ezberdintasun honek baditu hainbat arrazoi, portaera soziala geografikoki aldatzea esaterako. Hala ere, kausa nagusietako bat gaur egun erabiltzen diren eredu epidemiologikotan datza, hauek ez baitituzte gizarte portaeraren eta neurri ez-farmazeutikoen eragina era zuzenean aztertzeko aukera ematen.

Eredu epidemiologiko gehienek homogeneotasun baldintza onartzen duten ekuazio diferentzial sistemetan edota *ad hoc* ezarritako kontaktu ereduetan oinarritzen dira [6]. Eredu hauek gaixotasunen hedapena eskala handian aurrean ditzaketen arren, gizarte dinamika edo honengan eragiten dituzten neurrien eragina pandemiaren hedapenean auresatea ez da berehalakoa.

Ekuazio diferentzialetan oinarritutako metodo gehienak *Kermack-McKendrick* ereduan oinarritzen dira. Ekuazio diferentzial sistema hauek gaixotasun, baldintza geografiko edota portaera soziologiko ezberdinetara moldatzeko parametro edo funtzio ezberdinak doitzen dira, heriotza, suspertze eta kontaktu ratioak esaterako [7]. Azken hau, kontaktu ratioa, denbora unitateko ematen diren kontaktuei dagokio, eta zuzenean erlazionaturik dago kutsatze kopuru eta gaixotasunaren hedapen abiadurarekin.

Konfinamendu, ordutegi murrizketa edo mugen itxieren helburua kontaktu kopurua, eta ondorioz, kontaktu ratioa murriztea da. Hala ere, neurriek kontaktu ratioan eragingo duten aldaketa zehatza auresatea zaila da. Hau dela eta ekuazio diferentzial sistemen doikuntza neurrien eragina pairatu eta behatu ostean egin izan ohi da.

Lan honek epidemiak eskala txikiagoan aztertzekeo gai den metodo bat garatzea izango du helburu, neurri ez-farmazeutikoen eta portaera sozialaren eragina era sakonago batean aztertuz. *Java* programazio-lengoaian oinarrituz, gaixotasunen hedapenerako oinarrizkoak izango diren pertsona, toki eta gaixotasunei dagozkien klaseak eraikiko dira.

Oinarrizko klase hauek erabiliz biztanleriaren lagin esanguratsu bat eraikiko da, herri bat. Herri hau eraikitzekeo Oñati herriko datuak hartuko dira eredu gisa. Pertsona klasean oinarrituz eta datu demografikoak irizpide izanik biztanleria eraikiko da, eta herri mailako enplegu datuak erabiliz herria toki sinpleagoetan deskonposatuko da, etxe, eskola, enpresa edota tabernetan esaterako.

Espanian eta Euskal Autonomia Erkidegoan eginiko ikerketa eta inkesta soziologikoe-tan oinarrituz, portaera klase bat eraikiko da, pertsona bakoitzak aldiune bakoitzean toki ezberdinetan egoteko probabilitatea zehaztuko duena. Portaera klasean COVID-19 gaixotasunaren hedapena moteltzekeo ezarritako neurriak inplementatuko dira, konfinamendu, bilkuretakeo pertsona kopuru maximoaren murrizte eta tabernen itxiera besteak beste.

Programaren baliagarritasuna aztertzekeo, COVID-19 gaixotasuna emulatzen saiatuko da, egun ezagutzen diren datuetan oinarrituz. COVID-19 gaixotasunaren hedapenaren aurka hartu ziren neurriak era kronologikoan errepikatuz, 2020ko maiatzetik aurrera emandako normaltasunerako itzulia simulatuko da. Azkenik, konfinamendu eta gaixotasunaren detekzio testak garatuz aztarnari sistema bat ezarriko da. Programa ebaluatzekeo, lorturiko emaitzak Oñatin COVID-19ak izandako eboluzioarekin alderatuko dira.

Metodo honen garapenaren helburu nagusia portaera sozialak eta neurri ez farmazeutikoek epidemien hedapenean duten eragina simulatu eta aztertzen lagunduko duen tresna eraikitzea da, herri, auzo, fakultate edo lantokiak bezalako eremuetan segurtasun eta prebentzio neurri efektiboagoak diseinatzen lagundu dezakeen tresna eraikitzea izanik xede.

1. Kapituluia

Oinarri teorikoak

Garatu nahi den metodoak gizabanakoen arteko transmisioan oinarrituko denez, lehen pausoa gaixotasunaren transmisio probabilitatea definitzea izango da. Programa transmisio metodo ezberdinak inplementatzeko gai izan dadin, patogenoen transmisio metodoak aztertu eta hauek deskribatzen dituen eredu matematiko bat garatuko da.

Programak gaixotasun ezberdinak simulatu dezan, transmisio metodoa parametro generikoak erabiliz adieraziko da, ondoren, gaixotasunen eragile diren patogenoen transmisio metodoaren arabera, parametro hauei balio egokiak esleituko zaizkie.

Portaera sozialak transmisio probabilitatean duen eragina era egokian simulatzeko, ezinbestekoa izango da ekuazioetan pertsonen arteko distantzia eta gaixoekiko esposizio denbora kontuan izatea. Hau dela eta, kutsatze probabilitatea kalkulatzeko lehen pausoa pertsona batek patogeno batekiko izan duen esposizioa kalkulatzeko izango da. Amaitzeko, jasaniko esposizioaren arabera kutsatze probabilitatea kalkulatu da.

1.1 Transmisio metodo nagusiak

Patogenoak era ezberdinetan hedatu daitezke. Transmisioa pertsona batetik bestera zuzenean ematen denean bi transmisio mota sailkatu ohi dira. Bide zuzenenak kontaktu bidezko eta *Flügge* tanten bidezko transmisioak dira. Bide ez-zuzenekoak berriz, aireko partikulen bidez, zeharkako kontaktuz, hau da, patogenoak dituzten objektuak (fomiteak) ukituz eta bide fekal-oralez ematen diren transmisioak dira. Amaitzeko zeharkako transmisio bideak ditugu, non patogenoa beste organismo baten bitartez transmititzen den. Azken hauek bitarteko ostalariak edo gaixotasun bektoreak izan daitezke, txakurrak amorruren kasuan eta eltxoak malariaren kasuan esaterako.

Garapen honetan, portaera sozialarekin erlazio estuenak dituzten transmisio metodoak aukeratuko dira, hau da, distantziamendu eta esposizio denbora bezalako faktoreekiko menpekotasun handiena dituztenak, *Flügge* tanten bidezko, aire partikulen bidezko, kontaktu zuzeneko eta zeharkako kontaktuko transmisioak hain zuzen.

1.1.1 *Flügge* tanten bidezko transmisioa

Hitz egitean, eztul edota doministiku egitean kanporatzen diren listu partikulak dira, gehienbat urez osatuta daudenak. Hazizurria (parotiditisa) eragiten duen birusaren transmisio bide nagusietako bat, kontaktu zuzenarekin batera, tanta hauek dira.

Flügge tanten helmena lurruntze eta erorketak mugatzen dute. Tamaina ezberdineko tantengan fenomeno hauek duten eragina azaltzen duen kurbari *Wells* kurba deritza [8].

Tanten hedapenean haize, tenperatura, hezetasun eta hedapena ematen den eremuaren geometriaren arabera aldatu daiteke [9], hala ere, hedapenaren eredu sinplifikatu bat garatzeko, hedapenak distantzia eta denborarekiko duen portaera asintotikoa hartuko da kontuan. Kutsatze probabilitatea kalkulatzeko, pertsona kutsatu bat *Flügge* tanta igorle gisa kontsideratuko da, eta jario honetatik inguruneke patogeno dentsitatea (ρ) kalkulatu-ko da. Lehen hurbilpen baten, tanta hauek era isotropo eta homogeanoan hedatzen direla kontsideratuko da, ondorioz airean dagoen patogeno dentsitatea pertsona kutsatuarekiko r distantziaren karratuarekiko alderantziz proportzionala izango da, $\rho_{FT} \propto r^{-2}$.

Bestalde, tanten lurruntze eta erorketa prozesuak Poissonen prozesuak direla onartuz, tanten helmek banaketa esponentziala jarraitu beharko du, $\rho_{FT} \propto e^{-r/d_0}$ eran, non d_0 tanten helmekaren itxarondako balioa den. Hau dela eta, *Flügge* tantei dagokion patogeno dentsitateak distantziarekiko ondorengo menpekotasuna duela kontsideratu daiteke:

$$\rho_{FT} \propto \frac{e^{-r/d_0}}{r^2} \quad (1.1)$$

1.1.2 Aireko partikulen bidezko transmisioa

Flügge tantez gain, patogenoak aireko partikulen bidez hedatu daitezke. Tanten eta partikula hauen arteko ezberdintasun nagusiak tamaina eta partikulak osatzen duten materiala dira, orokorrean txikiagoak eta solidoak izan ohi dira eta. Partikula hauen bidez transmititzen diren patogenoen adibideak barizela eta elgorria eragiten dituzten birusak dira, besteak beste.

Partikula hauek txikiagoak izatearen ondorioz, erorketa askoz beranduago jasan ohi dute. Bestalde, solidoak edo partzialki urez osaturik daudenez, partikulen lurruntzeak ez du patogenoen hedapena mugatzen.

Berriro ere, pertsona kutsatua partikula igorle gisa kontsideratuz, eta erorketa eta lurrunketa prozesuak arbuiatuz, aireko partikula jarioari dagokion patogeno dentsitatea distantziaren karratuarekiko alderantziz proportzionala dela kontsideratu daiteke:

$$\rho_{AP} \propto \frac{1}{r^2} \quad (1.2)$$

1.1.3 Kontaktu zuzeneko transmisioa

Gaixotasun asko zuzeneko kontaktu bidez transmititu daitezke, sexu-transmisiozko gaixotasunak (STG-ak) edo mononukleosia adibidez. Zuzeneko kontaktua simulatzeko r_{min} atari distantzia bat ezarriko da. Kutsatutako eta kutsatu gabeko bi pertsonen arteko distantzia r_{min} baino txikiagoa izanez gero, kontaktua eman dela suposatuko da. Beraz, r_{min} baino distantzia txikiagoetarako balio finko bat duen eta r_{min} baino distantzia handiagoetarako nulua den maila funtzio bat erabiliko da kontaktuarekin erlazionatutako patogeno dentsitatea irudikatzeko.

$$\rho_K \propto \theta(r_{min} - r) \quad (1.3)$$

Non $\theta(r_{min} - r)$ adierazpena r_{min} balioan zentratutako Heaviside funtzioari dagokion.

1.1.4 Zeharkako kontaktu bidezko transmisioa

Zeharkako kontaktu bidezko transmisioa fomite bitartez ematen da, hau da, patogenoz kutsatuta dauden objektuak ukituz. Objektu hauek mota askotakoak izan daitezke, ura, janaria, pareta edo mahaitako gainazalak edo maiz ukitzen diren ingurunezko tresnak, ateetako heldulekuak adibidez.

Patogeno dentsitatearen ekuazioan zeharkako kontaktuari dagokion osagaia gehitu ahal izateko, objektu hauek ingurunean homogeneoki banatuta daudela onartuko da. Fomiteak pertsona kutsatua dagoen ingurunearen azalera (A) era homogeneoan banatuta badaude, patogeno dentsitatea konstantea eta ingurunearen azalerarekiko alderantziz proportzionala dela onartu daiteke:

$$\rho_{ZK} \propto \frac{1}{A} \quad (1.4)$$

1.2 Patogeno dentsitate eta esposizioa

Orain arte eginikoa hedapen mota bakoitzarentzat patogeno dentsitateen profilen portaera asintotiko hurbildua ondorioztatzea izan da. Hurrengo pausoa, adierazpen hauetatik abiatuz, hedapen mota hauek erabiliz transmititzen den edozein patogenori egokitu daitekeen adierazpen bat garatzea izango da.

Adierazpen hauek linealki konbinatuz, pertsona kutsatu bati dagokion patogeno dentsitate profil orokor bat ondorengo ekuazioarekin adierazi daiteke:

$$\rho = \alpha_{FT}\rho_{FT} + \alpha_{AP}\rho_{AP} + \alpha_K\rho_K + \alpha_{ZK}\rho_{ZK} \quad (1.5)$$

Non α aldagai erreal positiboak hedapenaren modulazio parametroak diren. Hauek aldatuz ekuazioa edozein gaixotasunera doitu daiteke. Adibidez, simulatu nahi den gaixotasunaren transmisio bide nagusia kontaktu bidezkoa baldin bada, gainontzeko osagaiei dagozkien α parametroei balio nuluak ezarriz lortuko litzateke gaixotasuna sortzen duen patogenoaren dentsitate profila.

Horretaz gain, patogenoen hedapena baldintza espezifikotara doitzeko erabili daitezke. Esaterako, maskara eraman ezean *Flügge* tanten eta aireko partikulen bidez transmititutako patogeno kopurua bikoizten bada, programan maskara eraman ezean balio hauek bikoizten duen baldintza implementatu daiteke. Modulatzaila hauek balio nulu edo positiboak har ditzakete.

Kutsatu bakoitzari dagokion patogeno dentsitate profila definituta, hurrengo pausoa k pertsona kutsatuak t aldiunean $r_k[t]$ distantziara dagoen pertsona osasuntsu bati helaraziko dion patogeno kopurua ($\varrho_k[t]$) definitzea izango da. Bi pertsona aldiune batean toki berean egon ezin daitezkeenez, pertsonen arteko distantziari balio minimo bat ezarri behar zaio, honela (1.1) eta (1.2) ekuazioetako dibergentzien arazoa zuzenduz. Balio minimo hau (1.3) ekuazioan definitutako r_{min} kontaktu distantzia izango da hain zuzen. Adierazpena laburtzeko $r = r_k[t]$ sinplifikazioa eginez, helarazitako patogeno kopurua ondorengo izango da:

$$\varrho_k[t] = \begin{cases} \alpha_{FT} \frac{e^{-r_{min}/d_0}}{r_{min}^2} + \alpha_{AP} \frac{1}{r_{min}^2} + \alpha_K + \alpha_{ZK} \frac{1}{A} & r \leq r_{min} \\ \alpha_{FT} \frac{e^{-r/d_0}}{r^2} + \alpha_{AP} \frac{1}{r^2} + \alpha_{ZK} \frac{1}{A} & r > r_{min} \end{cases} \quad (1.6)$$

Ekuazioko parametroek ez dute zertan finkoak edo konstanteak izan behar. Hainbat gaixotasunentzat kutsagarritasuna gaixotasunaren etaparen araberakoa izan oi da [11]. Aireko partikulen eta *Flügge* tanten bidezko transmisioen kasuan, egurats baldintzak, hezetan, tenperatura eta haizeak adibidez, edo maskara jantzita eramateak, patogenoen hedapenean eta hauen helmenean eragin dezake. Hau dela, α_i hedapen modulatzaila eta d_0 helmenearen itxarondako balioarentzat, adin, pertsona kutsatuen egoera eta ingurune baldintzekiko menpekotasuna ezarriko da.

Hurrengo pausua denboraren eragina implementatzea izango da. (1.6) ekuazioan definitutako dena k pertsona kutsatuak pertsona osasuntsuari t aldiunean helaraziko dion patogeno kopurua izan da. Hala ere, pertsona osasuntsuak t aldiunean ikusiko duen eta k pertsonatik datorkion patogeno kopurua ($P_k[t]$), ez da soilik aldiune horretan iritsitako patogeno kopurua izango, aurreko unean helarazitako patogenoen hondarra ere kontuan izan behar da.

Pertsona kutsatua patogenoak era uniformearen kanporatzen dituela eta hauek gorputzetik kanpo periodo bat biziraun dezaketela onartuko da. Erdibizitza τ izanik eta patogenoen heriotza Poissonen prozesua dela onartuz, ekuazioan $e^{-t/\tau}$ faktorea gehitu beharko da.

Programan denbora diskretizatuta izango denez, Δt denbora pausua izanik, pertsona osasuntsuak t aldiunean ikusiko duen eta k pertsona kutsatuak helarazitako patogeno kopurua ($P_k[t]$) t aldiunean k pertsonak helarazi dion patogeno kopuruaren ($\varrho_k[t]$ -ren) eta aurreko aldiunean ($t - 1$ aldiunean) posizio horretan ikusiko lukeen patogeno kopurua eta $e^{-\Delta t/\tau}$ faktorearen arteko biderkaduraren arteko batura izango da:

$$P_k[t] = \varrho_k[t] + P_k[t - 1]e^{-\Delta t/\tau} \quad (1.7)$$

Aurreko ekuazioan pertsona osasuntsuak t aldiunean ikusiko duen eta k pertsona kutsatuak helarazitako patogeno kopurua definitu da. Hainbat pertsona kutsatu dauden ingurune baten pertsona kutsatu guztiei dagozkien patogeno kopuruak batuz, pertsona osasuntsu batek t aldiunean izango duen esposizioa ($Q[t]$) lortu daiteke:

$$Q[t] = \sum_k P_k[t] \quad (1.8)$$

Pertsona osasuntsua epe luze bat egon bada kutsaturiko pertsonen gertutasunean, esposizio osoa (Q) denborari dagokion batukari bat gehituz kalkulatu beharko da:

$$Q = \sum_{t=h}^a Q[t] \quad (1.9)$$

Non h hasierako esposizio unea den, eta a amaierakoa.

Garapen honetan pertsona osasuntsu batek jasaten duen esposizioaren adierazpen hurbildu bat lortu da, hau da, pertsona batek gaixotasun generiko bat sortzen duten patogenoekin duen kontaktua esposizio denbora eta pertsona kutsatuekiko mantendutako distantziaren arabera. Hurrengo pausua, esposizio osoan oinarrituz, kutsatzeko probabilitatea kalkulatzeko izango da.

1.3 Kutsatze probabilitatea

Egun erabiltzen diren eredu epidemiologiko asko eredu konpartimentaletan oinarritzen dira. Eredu hauek populazioa talde edo “konpartimentu” ezberdinetan sailkatzen dute, *susceptible* edo S, *exposed* edo E, *infectious* edo I eta *recovered* edo R esaterako.

Gaixotasunei eboluzio ezberdinak ezarriz eta ekuazio diferentzial sistemak egokituz, eredu ezberdinak garatzen dira. Esaterako *susceptible* \rightarrow *infectious* \rightarrow *recovered* SIR ereduak izango litzateke eta periodo luzeko edo bizi osorako immunizazioa duten gaixotasunentzat balioko luke, adibidez, barizela. Eredu konplexuagoak ere garatu daitezke, *susceptible* \rightarrow *infectious* \rightarrow *recovered* \rightarrow *susceptible* edo SIRS esaterako, immunizazio epe motzagoko gaixotasunentzat, hotzeriaren kasua adibidez.

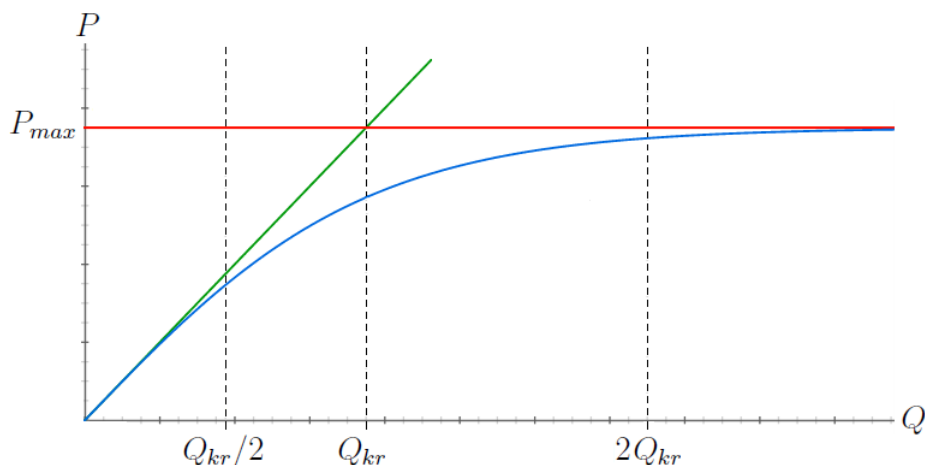
Konpartimentu hauek erlazionatzen dituzten parametroak trantsizio ratioak dira, eta pertsona batek konpartimentu batetik bestera igartzeko duen probabilitatea adierazten dute. Beste hainbat parametrotan oinarritzen dira, kutsatze ratioan esaterako. Era berean kutsatze ratioa gaixotasunaren kutsagarritasun eta kontaktu ratiotik ondorioztatzen da. Hala ere, balio hauek finkoak izan ohi dira, distantzia eta esposizio denbora kontuan hartzeko, aurreko atalean definitutako esposizio osoaren arabera kutsatze probabilitate bat definitu behar da.

Pertsona batek jasandako esposizio osoa (1.9) ekuazioan definitu delarik, hurrengo pausoa lortutako adierazpenetik abiatuz pertsona batek izango duen kutsatze probabilitatea definitzea izango da.

Patogeno dentsitate baxuentzat, kutsatze probabilitatea patogeno dentsitatearekiko proportzionala dela onartuko da, hau da, kontaktuan izandako patogeno kopurua bikoiztuz, patogeno hauetako bat gorputzera sartu eta pertsona infektatzeko probabilitatea bikoiztu egingo dela suposatuko da.

Hala ere, esposizio kritiko (Q_{kr}) batetik aurrera, probabilitateak balio maximo bat (P_{max}) izan beharko du, hau da, balio kritikotik aurrera patogenoa gorputzera sartzen dela onartuko da. Joera hauek kontuan izanik, kutsatze probabilitatea honela definituko da:

$$P = P_{max} \cdot \arctan\left(\frac{Q}{Q_{kr}}\right) \quad (1.10)$$



1.1. Irudia: Urdinez: Kontaktuan jarritako patogeno dentsitate eta kutsatze probabilitatearen arteko erlazioa. **Berdez:** $P = P_{max} \cdot Q/Q_{kr}$ erlazio lineala. **Gorritz:** P_{max} Kutsatze probabilitate maximoa.

Probabilitate maximoari eta patogeno dentsitate kritikoari dagozkien parametroek, (1.6) ekuazioko parametroek bezala, gaixotasunen arabera izateaz gain, ez dute zertan finkoak izan pertsona guztientzat. Probabilitateak osasun egoeraren arabera doituz, eredu konpartimentaletan oinarritutako ereduak koherentea den eredu bat garatu daiteke.

Bestalde, gaixotasunak adinarekiko duen menpekotasuna aztertuz eta parametro hauek adinaren arabera doitzuz, adin espezifikoko pertsonak egon ohi diren tokiak, eskola edo zahar etxeak esaterako, era zehatzagoan aztertu daitezke.

Immunizazio egoera ezberdinak ere aztertu daitezke. COVID-19 gaixotasunaren kasuan esaterako, non ustezko gaixotasun osteko immunizazio epean birkutsatze kasuak eman diren [12] edo txertoak %100eko eraginkortasunik ez duen, kutsatze probabilitatea doitu daiteke P_{max} probabilitate maximoa balio egokietara murriztuz.

2. Kapituluia

Programaren eraikuntza

Behin gizabanakoetan oinarritzen den kutsatzearen eredu matematikoa garatuta, hurrengo pausoa programaren eraikuntzarekin hasia izango da.

Simulazioentzako programa *Java* programazio-lengoaian eraikiko da. Atal honetan erabiliko diren klaseak eta hauen propietateak azalduko dira. Klaseak bi taldetan banatuko dira, klase generikoak eta klase espezifikoak. Programan dokumentazio sakonago bat eskaintzen den arren, programaren funtzionamendua hobeto ulertu ahal izateko kapitulu honetan klase nagusien funtsezko atributu eta metodoak azalduko dira.

Lan honen helburua toki eta esparru ezberdinetara egokitu eta hauek simulatzeko gai den tresna bat eraikitzea denez, kapitulu honen lehen atalean unibertsalak izango diren eta edozein simulazioentzat balio dezaketen klaseak eraikiko dira, oinarritzko tresnak izango direnak. Klase hauek gaixotasunei, pertsoneri edo tokiei dagozkien oinarritzko klaseak izango dira eta lehen atalean hauen atributu eta metodoetan sakonduko da.

Kapituluaren bigarren atalean, lehen atalean azalduko diren oinarritzko tresnak erabiliz, egitura konplexuagoak eraikiko dira, biztanleria eta tokiak. Klase hauek pertsona eta toki sinplez osatuta egongo dira eta *array* edo matrize erako datu egituretan antolatuko dira, eta pertsona eta toki partikularrak identifikatzeko *id* eta *pointer* gisa izendatuko diren indizeak erabiliko dira. Oñati herriko datuetan oinarrituz, simulaziorako espezifikoak izango diren biztanleria, toki distribuzio eta portaera sozialari dagozkien klaseak eraikiko dira.

2.1 Klase generikoak

Klase hauek programako oinarritzko elementuak izango dira, burutuko den Oñati herriko simulazioaz gain, simulatu nahi den beste edozein toki edo esparru simulatzeko baliagarri izan daitezke.

Oinarritzko klase hauek pertsona, toki eta gaixotasun klaseak izango dira. Klase hauek gain, pertsonen lista gisa balioko digun klase bat eraikiko da, toki bakoitzeko edo aztarnari sistemen listetako pertsonen jarraipen bat egitea ahalbidetuko duena.

2.1.1 *Person* klasea

Simulazioko pertsona bati dagokion klasea da. Pertsona baten portaeran oinarrizko izango diren atributuak adina, lantokia, bizitokia, familia taldea, “kuadrilla” edo talde soziala izango dira. Bestalde, gaixotasunaren propagazioarekin erlazionatutako atributuak osasun egoera, detekzio egoera eta konfinamendu egoera izango dira.

Pertsona batek aldiune bakoitzean kokapen (*Place*), posizio eta osasun egoera bakarria izango du, pertsona klaseko metodoak pertsonen oinarrizko ekintzak emulatuko dituzte, toki batera joan, posizio bat finkatu edota osasun egoera bat ezarri besteak beste.

Atributuak

Pertsonen atributuak hauen propietate eta egoerari buruzko informazioa adierazten dute. Garatutako eredu matematikoan oinarrituz, gaixotasunaren hedapenean eragin dezaketen ezaugarriak hartu dira kontuan soilik, hau da, pertsonen osasun egoeran edo gizarte portaeran eragin ditzaketen atributuak.

2.1. Taula: Pertsona klasearen oinarrizko atributuen taula.

Atributua	Datu mota	Balio posibleak	Azalpena
<i>age</i>	<code>int</code>	$0 \leq age$	Pertsonaren adina.
<i>home</i>	<code>int[2]¹</code>	(-)	Pertsonaren etxea adierazten duen <i>pointer</i> indizea. Indize honek etxea <i>Town</i> klasean identifikatzen du.
<i>workplace</i>	<code>int[3]¹</code>	(-)	Pertsonaren lantokia adierazten duen <i>pointer</i> indizea. Indizeak lantokia <i>Town</i> klasean identifikatzen du.
<i>currentplace</i>	<code>int[3]¹</code>	(-)	Pertsona non dagoen adierazten duen <i>pointer</i> indizea. Indizeak tokia <i>Town</i> klasean identifikatzen du.
<i>x, y</i>	<code>double</code>	$0 \leq x \leq xsize$ $0 \leq y \leq ysize$	Pertsonaren posizioa adierazten duten balioak.
<i>state</i>	<code>string</code>	uninfected, latent, incubation, asymptomatic, symptomatic, hospitalized, recovered, vaccinated, deceased	Osasun egoera, gaixotasunaren fase edo immunizazio egoeraren araberakoa. Egoerak iragankorrak edo zehaztugabeko luzerakoak izan daitezke.
<i>countdown</i>	<code>int</code>	$0 \leq countdown$	Osasun egoera iragankor bat duen pertsona baten kasuan, hurrengo osasun egoerara pasatzeko falta zaion egun kopurua.

Atributua	Datu mota	Balio posibleak	Azalpena
<i>hnumber</i>	int	$0 \leq hnumber$	Pertsona ospitaleratuak identifikatzen duen balioa. Ospitale gelak esleitzeko erabili daiteke.
<i>detected</i>	boolean	<i>true</i> <i>false</i>	Gaixotasunaren detekzio egoera, pertsona detektatua izan den edo ez adierazten duen boolearra.
<i>confined</i>	boolean	<i>true</i> <i>false</i>	Pertsona konfinatua dagoen edo ez adierazten duen boolearra.
<i>masked</i>	boolean	<i>true</i> <i>false</i>	Pertsona batek maskara jantzita duen edo ez adierazten duen boolearra.
<i>familygroup</i>	int	$0 \leq familygroup$	Familia taldea identifikatzen duen zenbakia. Etxe bereko pertsonak familia bat osatzen dute, familia talde bat hainbat familiek osatutako amalgama bat da.
<i>socialgroup</i>	int	$0 \leq socialgroup$	Pertsonaren talde soziala identifikatzeko zenbakia

Metodoak

Pertsona klasearen oinarrizko atributuak definiturik, hurrengo pausoa metodoak definitzea izango da. Metodo nagusiak `setAttribute(value)` motakoak izango dira, hau da, atributu bakoitzari desiratutako balioa esleitzen dieten metodoak.

Metodo hauetaz gain test batek eta egun baten igarotzeak pertsona batengan duten eragina emulatzen duten `test()` eta `nextday()` metodoak eraikiko dira.

2.2. Taula: Pertsona klasearen oinarrizko metodoen taula.

Metodoa	Argumentuak (Datu mota)	Itzulitako balioa	Azalpena
<code>test()</code>	(-)	(-)	Pertsona batengan test batek duen eragina simulatzen duen metodoa. Positiboen kasuan pertsona <i>detektatua</i> izatera pasako da. Positibo faltsu eta eraginkortasun tasak ere zehaztu daitezke.
<code>nextday()</code>	(-)	(-)	Egun baten igarotzeak duen eragina simulatzen duen metodoa. <i>Countdown</i> balioa unitate baten murrizten da. Balioa 0-ra iristean gaixotasunaren hurrengo fase- ra igarotzen da <code>nextstage()</code> metodoa erabiliz.

¹`Town` klasean tokiak matrize formatuko datu egituratan antolatzen direnez, edozein toki `int`, `int[2]` edo `int[3]` motako *pointer*-ekin zehaztu daiteke, tokiak toki-matrizetan duen posizioa adieraziz.

2.1.2 *Peoplelist* klasea

Gizabanakoetan oinarritutako simulazioak egiteko gai den programa eraiki nahi denez, pertsona eta toki kopurua handitu ahala kostu konputazionala haztea espero daiteke. Hala ere, toki batean dagoen jende kopurua lortu nahi den bakoitzean simulazioko pertsona guztien *currentplace* atributua banan banan aztertu behar bada, eta toki guztientzat ekorketa bera egin behar bada, programak praktikotasuna galduko du simulaturiko eremua handitu ahala.

Arazo hau konpontzeko, pertsonen lista gisa balioko duen klase bat eraikiko da. Lista hauek toki bakoitzeko pertsonak era praktikoko baten eskuragarri izateaz gain, gaixotasunaren aztarnatze sistemak inplementatzeko lagungarriak izan daitezke.

Atributuak

Peoplelist klasearen atributuak azaltzen hasi aurretik, *Population* klasean *pertsonak* nola antolatuta egongo diren azaldu behar da. Tokiekin egingo den bezala, pertsona kopuru handiak errazago maneiatzeko pertsonak *array* edo matrize motako datu egituretan antolatuko dira. Datu egitura hauen arabera, pertsona bat identifikatzeko balio edo balio sorta bat beharko da. *Array* edo lista erako egituren kasuan balio bakarra beharko da, pertsonak zerrendan duen posizioa adierazten duena. Matrize erako egituretan berriz bi balio beharko dira, errenkada eta zutabea adierazten dutenak. Datu egitura konplexuak erabili nahi badira, balio gehiago beharko dira.

Pertsona bat identifikatzeko beharko den balio kopurua *idsize* izango da. Pertsona bat identifikatzeko balioak *idsize* tamainako *array*-etan sartuko dira. *Array* hauek pertsonen *id* edo identifikatzaileak izango dira. *Peoplelist* klaseak bi atributu izango ditu, *idsize* eta pertsonen identifikatzaileen zerrenda izango den *people array*-a.

2.3. Taula: Pertsonen lista klaseari dagokion oinarritzko atributuen taula.

Atributua	Datu mota	Balio posibleak	Azalpena
<i>idsize</i>	<code>int</code> ²	$1 \leq idsize$	Pertsona baten identifikatzaileak izango dituen balio kopurua; <i>Population</i> klasean pertsonen egiturak izango dituen dimentsio kopurua.
<i>people</i>	<code>int[]</code> [<i>idsize</i>] ²	(-)	Pertsonen identifikatzailez osatutako <i>array</i> -a. Identifikatzaile kopurua ez da finkoa, pertsonak atera edo gehitzean aldatzen da.

²*Population* klasean pertsonak matrize erako datu egituratan antolatzen direnez, edozein pertsona `int[idsize]` motako *id*-arekin zehaztu daiteke, pertsonak pertsona-matritzetan duen posizioa adieraziz.

Metodoak

Peoplelist klasean pertsonekin eragiketak egin ahal ditzaketen oinarrizko metodoak definituko dira, pertsonak gehitzea eta kentzea esaterako.

2.4. Taula: Pertsonen lista klaseari dagokion oinarrizko metodoen taula.

Metodoa	Argumentuak (Datu mota)	Itzulitako balioa	Azalpena
<code>addperson()</code>	<i>id</i> (<code>int</code> [<i>idsize</i>])	(-)	<i>People</i> atributoan gehitu nahi den pertsonaren identifikatzailea eranstu du.
<code>removeperson()</code>	<i>id</i> (<code>int</code> [<i>idsize</i>])	(-)	<i>People</i> atribututik kendu nahi den pertsonaren identifikatzailea ezabatzen du.
<code>emptylist()</code>	(-)	(-)	<i>People</i> atribututik pertsona guztien identifikatzaileak ezabatzen ditu.
<code>isrelevant()</code>	(-)	<code>boolean</code>	<i>Toki</i> bat simulaziorako esanguragarria izango den edo ez adierazten du. <i>Toki</i> bat esanguragarria izan dadin, <i>peoplelist</i> -ean gutxienez pertsona osasuntsu batek egon beharko du, eta hau kutsatzeko probabilitatea ezin da nulua izan.
<code>isin()</code>	<i>id</i> (<code>int</code> [<i>idsize</i>])	<code>boolean</code>	<i>id</i> identitatea duen pertsona zerrenda barruan dagoen edo ez adierazten duen metodoa.

2.1.3 *Place* klasea

Simulazioko oinarrizko toki fisiko bati dagokion klasea izango da. Atributuei dagokionez, bi mota nagusitan sailkatuko dira. Alde batetik tokiaren ezaugarri fisikoei dagozkien atributuak definituko dira. Bestetik, tokian aurkitu daitezken pertsonekin erlazioatutako atributuak eraikiko dira.

Atributuak banatzeko erabili den irizpidearekin jarraituz, metodoak ere bi taldetan banatuko dira. Ezaugarri fisikoei dagokionez, hauek ezarri eta aldatzea ahalbidetuko diguten metodoak eraikiko dira. Tokiko pertsonen dagokionez berriz, tokian pertsonak sartu edo ateratzeko metodoak definituko dira.

Atributuak

Pertsonen kutsatzean eragin dezaketen faktoreek osatuko dute, tokiko ezaugarri fisikoek eta tokian aurkitzen diren pertsonen hain zuzen.

2.5. Taula: Toki klaseari dagokion oinarriko atributuen taula.

Atributua	Datu mota	Balio posibleak	Azalpena
<i>xsize, ysize</i>	double	$0 < xsize$ $0 < ysize$	Zabalera eta luzera dimentsioak adierazten dituzten balioak.
<i>temperature</i>	double	(-)	Tenperatura adierazten duen balioa.
<i>humidity</i>	double	$0 \leq humidity \leq 100$	Hezetasun maila (%) adierazten duen balioa.
<i>wind</i>	double	(-)	Haizearen abiadura adierazten duen balioa.
<i>presentpeople</i>	<i>Peoplelist</i>	(-)	Tokian dauden pertsonak adierazten duen pertsonen lista.

Metodoak

Alde batetik, tokiaren ezaugarri fisikoak ezarri eta aldatzeko `setAttribute(value)` erako metodoak eraikiko dira, atributu bakoitzari azaldutako balioa ezartzen dietenak.

Bestetik, tokiko pertsonak maneiatzea ahalbidetzen duten metodoak eraikitzeko, *Peoplelist* klasera **2.4 Taulan** definitutako metodoak erabiliko dira, metodo hauek *presentpeople* atributura luzatuz. Luzapen honi esker tokira pertsonak gehitu, bertatik pertsonak atera edo tokia guztiz hustea ahalbidetzen duten metodoak eraikiko dira.

2.1.4 *Illness* klasea

Klase orokorrekin amaitzeko gaixotasunari dagokion klasea eraikitzea falta da. Gaixotasunaren transmisioa eta eboluzio edo garapena zehazten dituzten parametroek izango dira klasearen atributu nagusiak.

Klaseko metodoak berriz, epidemiaren hedapenean funtsezkoak diren kalkuluak egitea ahalbidetuko dituztenak izango dira, patogeno dentsitatea eta kutsatze probabilitateak kalkulatzeko balio dituztenak besteak beste.

Atributuak

Gaixotasunaren atributuak oinarri teorikoan definitutako parametroak izango dira. Alde batetik, patogenoen hedapen eta kutsatze modua doitzeko (1.6) eta (1.10) ekuazioetan definitutako parametroak eraikiko dira. Bestetik, **1.3 Kutsatze probabilitatea** atalean aipatutako transferentzia ratioak eraikiko dira, pertsona batek *osasun egoera* batetik bestera igarotzeko duen probabilitatea zehaztuko dituztenak.

2.6. Taula: Gaixotasun klasearen oinarriko atributuen taula.

Atributua	Datu mota	Balio posibleak	Azalpena
α_i	(-) ³ , double	$0 \leq \alpha_i$	Patogenoen hedapen modulatzailleak. Gaixotasuna sortzen duen patogenoaren hedatze modua modulatzten dituzten parametroak.
r_{min}	double	$0 \leq r_{min}$	Kontaktu fisikoa egon dadin bi pertsonen artean izan beharreko distantzia.
d_0	(-) ³ , double	$r_{min} \leq d_0$	<i>Flügge</i> tanten helmenaren itxarondako balioa.
P_{max}	(-) ³ , double	$\%0 \leq P_{max} \leq \%100$	Kutsatze probabilitate maximoa adierazten duen balioa ehunekotan adierazita.
Q_{kr}	(-) ³ , double	$0 \leq Q_{kr}$	Esposizio kritikoari dagokion balioa.
$rate_i$	(-) ³ , double	$\%0 \leq rate \leq \%100$	Hurrengo osasun egoera zein izango den determinatzen duen probabilitate edo ratioa.
$period_i$	int	$0 < period$	Gaixotasunak sorturiko osasun egoera iragankor ezberdinen iraupena zehazten dituzten balioak.

Kontaktu distantzia izan ezik, gainontzeko atributuen datu egitura ez da zehaztuta egongo. Datu egitura simulatu nahi den gaixotasunaren hedapen eta eboluzioan eragiten duten menpekotasunaren araberakoa izango da.

Esaterako, garatu nahi den programan **adin**, **temperatura** eta **haizearekiko** menpekotasunik ez izatea erabakitzen bada, nahikoa izango da **double** edo **int** erako atributuak ezarriz. Askatasun gradu bat sartzea erabakitzen bada, esaterako atributuei adinarekiko menpekotasuna ezarriz, pertsonen adinaren arabera balio egokiak itzuliko dizkiguten datu egiturak erabili beharko dira, `Map<age, double>` motako datu egitura edo hiztegiak adibidez. Askatasun gehiago sartu nahi izanez gero, eguratsaren baldintzak esaterako, datu mota konplexuagoak edo `function(age, temperature, wind, humidity)` erako funtzioak erabili beharko dira.

Atributu hauen egitura finkatzeak garatutako programa eta metodoaren moldagarritasuna murriztu dezakeenez, ez finkatzea eta erabiltzailearen esku uztea erabaki da.

Metodoak

Gainontzeko klasetan bezala, `setAttribute(value)` erako metodoak eraikiko dira, atribuentzat aukeratutako datu egituretara egokituz. Oinarri teorikoetan garatutako kutsatze ereduko kalkuluak burutzen dituzten metodoak ere eraikiko dira.

³ Erabilitako datu egitura mota gaixotasuna deskribatzeko erabilitako parametro hauek izango dituzten menpekotasunen arabera egokitu behar da.

2.7. Taula: Gaixotasun klaseko oinarrizko metodoen taula.

Metodoa	Argumentuak (Datu mota)	Itzulitako balioa	Azalpena
<code>pathogendensity()</code>	<i>healthyperson</i> (<i>Person</i>) <i>infectedperson</i> (<i>Person</i>)	double	Pertsona osasuntsu eta infektatu bana emanda, pertsona osasuntsuak ikusitako patogeno kopurua kalkulatzen du (1.6) ekuazioako adierazpenean oinarrituz.
<code>infectionprobability()</code>	<i>healthyperson</i> (<i>Person</i>) <i>exposure</i> (double)	double	<i>Healthyperson</i> pertsona osasuntsuak <i>exposure</i> esposizio osoa izanik, (1.10) ekuazioan oinarrituta, kutsatzeko izango duen probabilitatea kalkulatzen duen metodoa.
<code>nextstage()</code>	<i>infectedperson</i> (<i>Person</i>)	(-)	<i>Infectedperson</i> pertsona infektatua izanik, bere hurrengo <i>osasun egoera</i> zein izango den zehazten duen metodoa. Hurrengo egoerara <i>countdown</i> balioa 0-ra iristean pasatzen denez, metodo hau <code>nextday()</code> metodoaren barruan deituko da.

2.2 Klase espezifikoak

Orain arte eraikitako klaseak orokorrak izan dira eta egokitzapen gutxi batzuk eginez edozein eremu simulatzen duten programetan inplementatu daitezke. Hurrengo pausoa, klase orokor hauetaz baliatuz, herri bat simulatzeko baliagarri diren klase espezifikoak eraikitzea izango da.

Gizabanakoetan oinarritutako programa bat erabiliz epidemia baten hedapena herri mailan simulatu nahi izanez gero, herria bera baino eremu zabalago bat simulatu beharko litzateke. Herritik kanpo ikasi edo lan egiten duen herriko jendea eta herrian bertan ikasi edo lan egiten duen kanpoko jendea funtsezkoak izan daitezke inguruko herri eta herrian epidemiak izango duen hedapenean eta.

Hala ere, lan honen helburu nagusia neurri ez-farmazeutikoen eta portaera sozialaren eragina aztertzeko gai den metodo bat garatzea denez, soilik herri bakarra izango den eremu bat eraikiko da, inguruko herriekin egon daitezken elkarrekintzak arbuiatuz.

Herria ausaz ez eraikitzeko, eta erabilitako datuak kontrastagarriak izan daitezen, Oñati herriko datu demografikoak oinarritzat hartuko dira. Biztanleriaren portaera ahalik eta era zehatzenean simulatzeko berriz, estatu eta autonomia erkidego mailan aisialdiari buruz eginiko inkestak eta herriko enplegu datuak izango dira irizpide.

2.2.1 *Town* klasea

Herria osatuko duten eremu fisikoak eraikitzeko *Place* klasean erabiliko da. Place klasea erabiliz eraiki daitezken lekuak geometria laukizuzeneko toki sinpleak direnez, toki konplexuagoak eraikitzeko hainbat toki erabiliko dira.

Etxe bat simulatzeko adibidez 5~6 toki ezberdin erabiliko dira, toki bakoitza gela bat izanik. Kasu hau nahiko sinplea izan arren, eremu handiago eta konplexuagoak simulatzeko estrategia bera erabili daiteke. Eskola bat esaterako ikasgela eta bainugela edo korridoreak bezalako gune komunetan deskonposatu daiteke.

Eraikiko den herriko leku kopurua zehazteko Oñati herriko enplegu datuak hartuko dira erreferentzia gisa. **2.8 Taulan** Oñatin sektoreko dauden enpresa eta langile kopurua daude ikusgai. Datu hauetan oinarrituz simulazioan eraiki beharreko lan eremu mota eta kopurua ondorioztatuko dira.

2.8. Taula: Oñati herriko enplegu datuak, enpresa eta langile kopurua eta ehunekoak sektoreka adieraziz. *Iturria: Gizarte Segurantzako Institutu Nazionala, 2020-ko ekaina*

	Langileak		Enpresak	
	Kopurua	%	Kopurua	%
Nekazaritza			4	0.9
Industria	2976	60.7	60	13.7
Estraktiboa	3	0.1	1	0.2
Manufaktura-enpresa	2965	60.5	56	12.8
Energia-horniketa	8	0.2	3	0.7
Eraikuntza	157	3.2	15	3.4
Zerbitzuak	1736	35.4	360	82.0
Handizkako merkataritza, txikizkako merkataritza eta ibilgailuen konponketa	349	7.1	60	13.7
Garraio eta biltegiatzea	58	1.2	8	1.8
Ostalaritza	269	5.5	38	8.7
Informazio eta komunikazioa	53	1.1	2	0.5
Finantza edo aseguruak	74	1.5	5	1.1
Higiezinen jarduerak	10	0.2	3	0.7
Jarduera profesional, zientifiko eta teknikoak	171	3.5	17	3.9
Administrazio jarduerak	64	1.3	4	0.9
Administrazio publikoa eta defentsa	106	2.2	6	1.4
Hezkuntza	283	5.8	10	2.3
Osasun jarduerak eta gizarte zerbitzuak	53	1.1	10	2.3
Jarduera artistiko, ludiko eta entretenimendua	14	0.3	1	0.2
Bestelako zerbitzuak	105	2.1	19	4.3
Jarduerak etxebizitzetan	127	2.6	177	40.3
GUZTIRA	4903	100.0	439	100.0

Datu hauetan oinarrituz, lan eremuak sei esparru ezberdinetan sailkatuko dira.

Lehen eremu mota manufaktura-enpresa edo fabrika erako lantokiek osatuko dute, non bataz beste 50 pertsonatik gora dauden lanean enpresako. Txikiagoak diren eta 10-15 pertsona inguruko lantoki, bulego edo tailerrak bigarren talde batean sailkatuko dira.

Ondoren ostalaritza enpresak izango dira, kafetegi, hotel, taberna eta jatetxeak. Lau-garren taldea dende osatuko dute.

Azken bi taldeak irakaskuntza eta osasun zentroek osatuko dute. Oñatin ospitalerik egon ez arren, ospitaleratuak izan daitezken eta osasun arloan lan egiten duten pertsonak simulatu ahal izateko, ospitalea gisa funtzionatuko duen eremu bat gehituko da.

Lantokiez gain, bi toki mota gehituko dira. Alde batetik zonalde publikoak eraikiko dira, plaza, parke, kale edo kiroldegiak adibidez. Bestetik etxeak gehituko dira, non etxe kopuru zehatza *Population* klasean ondorioztatuko den.

Pointer: Aipaturiko toki mota bakoitzari $Place_{n,m}$ toki-matrize bat esleituko zaio. n Balioak eremu mota bakoitzarentzat zenbat eremu egongo diren zehaztuko du, zenbat etxe, denda edo enpresa egongo diren adibidez. Eremu hauetako bakoitza zenbat tokitan (gelatan, ikasgelatan...) banatuta dagoen berriz m balioak zehaztuko du. Banaketa honekin *Town* klasean toki bat zehazteko hiru balio beharko dira. Lehen k balioak tokiaren eremu mota identifikatuko du. Bigarren i balioak tokia zenbatgarren eremuan dagoen zehaztuko du ($0 < i \leq n$). Azken j balioak tokia eremua osatzen duten tokien artean identifikatuko du ($0 < j \leq m$). Toki bat identifikatzen duten (k, i, j) balio sorta hau **pointer** edo toki adierazle gisa izendatuko da.

Atributuak

Klasearen atributuak herriko eremu motak izango dira. Mota bakoitzeko atributu bat eraikiko da, zortzi guztira. Hauetako bakoitza toki-matrize batek osatuko du.

2.9. Taula: *Town* klaseari dagokion oinarritzko atributuen taula.

Atributua	Datu mota	Azalpena
<i>homes</i>	<i>Place</i> [3855] [4]	Etxeei dagozkien tokiak. 3855 etxe egongo dira eta bakoitza 4 toki edo gelak osatuko dute.
<i>bigfactories</i>	<i>Place</i> [56] [5]	Manufaktura-enpresei dagozkien eremuak. Enpresa bakoitza sektore edo produkzio-kate eta gune komunetan banatuta egongo da.
<i>smallbusiness</i>	<i>Place</i> [100] [5]	Enpresa txiki, tailer edo bulegoak. Hauetako bakoitza 5 toki ezberdinetan banatuko da.

Atributua	Datu mota	Azalpena
<i>schools</i>	<i>Place</i> [12] [13]	Eskola eta unibertsitateak. 3 eskola nagusi egongo dira, eskola bakoitza 3 adin tartetan banatuko da (0-5, 6-11 eta 12-17). Unibertsitatearekin banaketa berdina egingo da, guztira 12 eremu edo ikasketa zentro izanik. Belaunaldi bakoitza 6 ikasgela banatuta egongo da (2 eskolako), honela eskola bakoitza 12 ikasgela eta zona komunek osatuko dute (13 toki).
<i>hostelry</i>	<i>Place</i> [38] [11]	Jatetxe, taberna edo hotelak. Hauetako bakoitza zona komun nagusi batek eta 10 zona espezifiko eta txikiagok (gela edo mahaiek) osatuko dute.
<i>shops</i>	<i>Place</i> [50] [1]	Dendak. 50 denda ezberdin egongo dira, eta bakoitza toki bakar batek osatuko du.
<i>healthcenter</i>	<i>Place</i> [1] [26]	Ospitalea. Bakarra izango da, eta 25 banako gela eta zona komunek osatuko dute.
<i>publicplaces</i>	<i>Place</i> [1] [70]	Gainontzeko intereseko leku publikoak, kalea, parke, liburutegi, kiroldegi edota zahar egoitzako zona komunak, besteak beste.

Metodoak

Herriari dagokion klaseko metodoak eraikitzeke *peoplelist* klasearen metodoak erabiliko dira, *pointer* toki adierazlea erabiliz toki espezifikoak aukeratzeko.

2.10. Taula: *Town* klaseari dagokion oinarrizko metodoen taula.

Metodoa	Argumentuak (Datu mota)	Azalpena
<i>addperson()</i>	<i>id</i> (<i>int</i> [<i>idsize</i>]) <i>pointer</i> (<i>int</i> [3])	<i>Pointer</i> -ak adierazitako tokian <i>id</i> identitatea duen pertsona gehitzen du pertsonaren <i>id</i> -a tokiaren <i>presentpeople</i> atributuan gehituz.
<i>removeperson()</i>	<i>id</i> (<i>int</i> [<i>idsize</i>]) <i>pointer</i> (<i>int</i> [3])	<i>Pointer</i> -ak adierazitako tokitik <i>id</i> identitatea duen pertsona ateratzen du pertsonaren <i>id</i> -a tokiaren <i>presentpeople</i> atribututik kenduz.
<i>emptyplace()</i>	<i>pointer</i> (<i>int</i> [3])	<i>Pointer</i> -ak adierazitako tokia husten duen metodoa tokiaren <i>presentpeople</i> atribua hustuz .
<i>emptytown()</i>	(-)	Toki guztiak husten duen metodoa toki guztien <i>presentpeople</i> atributuak hustuz .

2.2.2 *Conduct* klasea

Behin herria eraikita, hurrengo pausua pertsonen portamoldea simulatzeko balioko duen *Conduct* klasea eraikitzea izango da. Klase honen helburua aldiune bakoitzean pertsona batek izango duen posizioa zehaztea izango da.

Simulazioan bi egun mota egongo dira, lanegunak eta jaiegunak, eta bakoitza zortzi orduko hiru tartetan banatuko da, lantokiko lan denbora, atsedeen denbora eta aisialdi denbora. Zortzi ordu hauek 30 minutuko **jardueratan** banatuta egongo dira. Simulazioko denbora pausoa ((1.7) ekuazioko Δt) minutu bat izango da.

Lanegunetan, lanorduei dagokion denbora tartean [manufaktura enpresa](#), [enpresa txiki edo bulego](#), [eskola](#) eta [osasun zentroetan](#) ikasi edo lan egiten duen jendea beraien [lantokietan](#) egongo dira. Jaiegunetan berriz denbora tarte hau etxean edo aisialdi ekintzak burutzen igaroko dute. Toki hauetan lan egiten ez duten pertsonak ere denbora tarte hau etxean edo aisialdi ekintzak burutzen igaroko dute.

Aisialdi denboran [denda](#) eta [ostalaritza](#) enpresetan lan egiten duten pertsonak beraien [lantokietan](#) egongo dira. Gainontzeko pertsonak aisialdi ekintzak burutuko dituzte. Aisialdi jarduerak banaka edo taldeka burutu ahal izango dira. Taldekako jardueratan [familia](#) edo [talde sozial](#) bereko kideekin jarduera bera egingo dute, kafetegi edo jatetxe bateko mahai berean bilkurak esaterako.

Amaitzeko, atsedeen denboran pertsona bakoitza bere [etxean](#) egongo da.

Salbuespenak pertsona [detektatuak](#), [konfinatuak](#) eta ospitaleratuak izango dira. Pertsona detektatuak egun osoa etxeko toki bakarrean isolamenduan egingo dute, etxeko zona komunetan egingo den jarduera bakarrean zehar izan ezik. Pertsona konfinatuek berriz egun osoa etxean egingo dute, etxeko gune komunetan egon ahal direlarik. [Osasun egoera](#) ospitaleratua duten pertsonak egun osoa ospitaleko gela baten egingo dute, gela [hnumber](#) balioaren arabera esleituko delarik.

Atributuak

Klasearen atributuak pertsonen helmuga zehazten lagunduko diguten balioek osatuko dute. Pertsona baten lantoki eta etxei buruzko informazioa bere atributuek ([workplace](#) eta [home](#)) adierazten digutenez, klase honen atributu gehienek aisialdi denboran buruturiko jarduerak zehazteko balioko duten datu egiturek osatuko dute.

Jarduera bat burutzeko probabilitatea `double` balio batekin adieraziko da, eta ausazko balio bat honekin konparatuz jarduera hau burutuko den edo ez determinatuko da. Pertsona batek jarduera ezberdinak egiteko duen probabilitate sorta `double[]` erako *array*-etan gordeko da, non balio bakoitza jarduera espezifiko bat burutzeko probabilitatea izango den.

2.11. Taula: *Conduct* klaseari dagokion oinarrizko atributuen taula.

Atributua	Datu mota	Azalpena
<i>profile</i>	Map <age, double[]>	Portaera profila. Jarduera bakoitza burutzeko probabilitatea biltzen duen <code>double</code> multzoa. Profil bakoitzean hainbat <code>double[]</code> probabilitate sorta izango ditugu, bakoitza adin tarte batera egokitua.
<i>maxpeople</i>	<code>int</code>	Taldeka burutzen diren familia eta kuadrilla jardueratan pertsona kopurua mugatzen duen balioa.

Euskal autonomia erkidegoan eta Espainian COVID-19 pandemia aurretik aisialdi denbora eta jardueri buruz eginiko inkesta soziologikoetan oinarrituz [13, 14], portaera profil estandar bat sortuko da. Profil hau erreferentzia gisa hartuz, neurrien eragina simulatzen duten profilak gehituko dira. Esaterako, ostalaritza enpresetan egoteko probabilitatea 0-ra murriztuz, ostalaritzaren itxiera simulatzen duen profila eraiki daiteke eta dendetan egoteko probabilitateari izan ezik gainontzeko balio guztiei balio nuluak ezarriz, konfinamendu egoera simulatu daiteke.

Profil hauetan oinarrituz eta bilkuretako pertsona kopuru maximoa aldatuz, pandemian zehar aplikatu diren neurriek portaera soziologikoan, eta ondorioz pandemiaren hedapenean izan duten eragina simulatuko da.

Metodoak

Klaseko metodoek pertsona batek burutu ditzaken jardueren tokia itzuliko dute. Horretaz gain, pertsonen posizioa eguneratzen duen metodo bat eraikiko da. Simulazioan denbora pausua (Δt) minutu bat denez, posizioak minuturo eguneratuko dira. Jarduerak berriz ordu-erdiro esleituko dira.

2.12. Taula: *Conduct* klaseari dagokion oinarrizko metodoen taula.

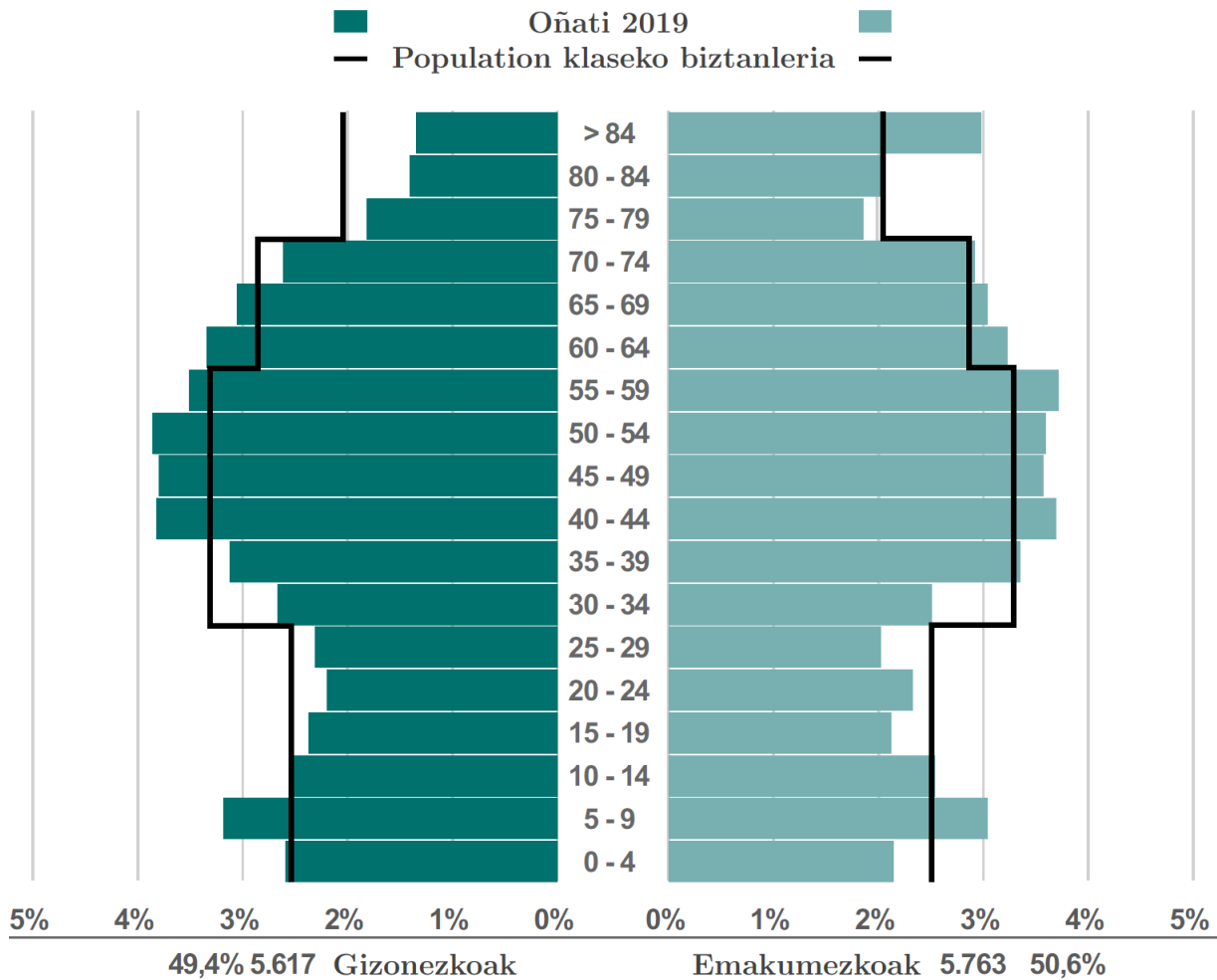
Metodoa	Argumentuak (Datu mota)	Itzulitako balioa	Azalpena
<code>getWTplace()</code>	<i>subject</i> (<i>Person</i>)	<i>pointer</i> (<code>int</code> [3])	Pertsona bat lan orduetan non egongo den itzultzen duen metodoa. Zona komunak dituzten lantoki kasuan jarduera bat zona komunetan egingo da. Gainontzekoak pertsonaren <i>workplace</i> -ean.
<code>getHTplace()</code>	<i>subject</i> (<i>Person</i>)	<i>pointer</i> (<code>int</code> [3])	Pertsona bat atseden denboran non egongo den itzultzen duen metodoa. Jarduera bat gutxienez etxeko zona komunetan burutuko da (baita pertsona detektatu eta konfinatuek ere).

Metodoa	Argumentuak (Datu mota)	Itzulitako balioa	Azalpena
<code>getLTplace()</code>	<i>subject</i> (<i>Person</i>)	<i>pointer</i> (<code>int [3]</code>)	Pertsona batek aisialdi denboran egingo duen banakako jardueraren kokapena itzultzen duen metodoa. Toki hau pertsonaren adin eta klase honen Portaera profilaren arabera aukeratzen da. Bi aldaera izango ditu, jaiegun eta lanegunetarako.
<code>getGroupP11()</code>	<i>subject</i> (<i>Person</i>)	<i>pointer</i> (<code>int [3]</code>)	Pertsona batek aisialdi denboran familiako kideekin egingo duen jardueraren kokapena itzultzen duen metodoa. Familia talde bakoitzari toki bat esleitzen zaio eta bertako kideek, asistentzia probabilitate eta pertsona kopuru maximoaren arabera jarduera burutuko dute.
<code>getGroupP12()</code>	<i>subject</i> (<i>Person</i>)	<i>pointer</i> (<code>int [3]</code>)	Pertsona batek aisialdi denboran kuadrilla bereko kideekin egingo duen jardueraren kokapena itzultzen duen metodoa. Talde sozial bakoitzari toki bat esleitzen zaio eta bertako kideek, asistentzia probabilitate eta pertsona kopuru maximoaren arabera jarduera burutuko dute.
<code>refreshxandy()</code>	<i>subject</i> (<i>Person</i>)	<i>x</i> (<code>double</code>) <i>y</i> (<code>double</code>)	Pertsonaren posizioa eguneratze duen metodoa. Posizioa ausaz, pertsona aurkitzen den tokiaren dimentsioen arabera, edo geometria partikular bat jarraituz esleitu daiteke, adibidez, ikasgela bateko mahai distribuzioa.

2.2.3 *Population* klasea

Herriko biztanleria eraikitzeke *Person* klasea hartuko da oinarritzat. Lehendabizi Oñatiko datu demografikoak irizpide izanik biztanleria adin ezberdinetan banatuko da. Biztanleria era koherentean simulatu ahal izateko, pertsona bakoitzari atributuak adinaren arabera ezarriko zaizkio, lagun talde edo lantokia esaterako.

Adin tarte bakoitzean dagoen biztanle kopuru zehatza inplementatuz gero, biztanleriari familia, talde sozialak eta lanpostuak ezartzeko, kopuru hauetara egokitzen diren metodo konplexuagoak diseinatu beharko lirarteke. Hau sinplifikatzeko, biztanleria adin tarte zabalagoetan banatu eta hauentzat batazbesteko biztanle kopurua kalkulatu da. 2019-an Oñatik zuen biztanleriaren adinaren arabera banaketa eta programan erabiliko den hurbilpenaren arteko konparaketa **2.1 Irudian** adierazi da:



2.1. Irudia: Errenkada berdez adierazita: Oñati herriko biztanleria adin tarteko emakumezko eta gizonezkoentzat 2019 urtean. Zuzen beltzez adierazita: Programan erabilitako biztanleriaren hurbilpena. *Iturria: Estatistikako Institutu Nazionala (INE).*

Id: Tokiak bezala, herriko 11380 biztanleak $Person_{n,m}$ pertsona-matrize irregular erako datu egituran antolatuko dira. Atributu asko adinaren arabera esleituko direnez, pertsonak adinaren arabera sailkatuko dira. Adin maximoa 94 urtetan ezarriz, 95 errenkada ezberdin izango ditu matrizeak, eta adin bakoitzeko pertsonak errenkadetan sailkatuko dira, adibidez $i = 37$ errenkadan 37 urteko pertsona guztiak sartuz. Bigarren j balioak pertsona bat adin bereko pertsonen artean bereiziko du. Errenkadako izango den pertsona kopurua ez da finkoa izango, **2.1 Irudian** ikusi daitekeen bezala. Pertsona bat identifikatzen duten (i, j) balioak pertsonaren identifikatzaile (**Id**) gisa izendatuko da.

Atributuak:

Population klaseko atributu nagusiak biztanleria adierazten duen pertsona-matrize irregular eta kutsatuen aztarnatzea ahalbidetuko duten [pertsonek listek](#) osatuko dute.

2.13. Taula: *Population* klaseari dagokion oinarrizko atributuen taula.

Atributua	Datu mota	Azalpena
<i>population</i>	<i>Person</i> [95] []	Pertsonen osatutako matrize irregular erako datu egitura. Adinaren arabera sailkatuta dagoen herriko biztanleria biltzen du.
<i>totestlist</i>	<i>Peoplelist</i>	Hurrengo egunean testa egin behar duten pertsonek lista . Kasu detektatu bat dagoen bakoitzeko, pertsonaren kuadrilla, familia edo lantokiko kideak listara gehituko dira.
<i>testinglist</i>	<i>Peoplelist</i>	Egunean bertan testa egingo zaien pertsonek lista . Eguna igarotzen den bakoitzean <i>totestlist</i> lista <i>testinglist</i> lista bilakatuko da, eta lehena hustu egingo da. Kasu detektatuak eta etxebizitza bereko pertsonak <i>observationlist</i> listara pasatuko dira, ez detektatuak berriz listatik kenduko dira.
<i>observationlist</i>	<i>Peoplelist</i>	Behaketan dauden pertsonek lista . Lista honetako pertsonak konfinatuta edo isolamenduan egongo dira. Gaixotasuna igaro ostean familiako kide guztiei bigarren test bat burutuko zaie, eta guztien emaitza negatiboa izanez gero listatik aterako dira.

Metodoak

Biztanleriaren portamoldean eragiten duten metodo gehienak klase honetan daude. Pertsonak testatu edo nonbaitera bidaltzen dituztenak izango dira metodo nagusiak:

2.14. Taula: *Population* klaseari dagokion oinarrizko metodoen taula.

Metodoa	Argumentuak (Datu mota)	Azalpena
<i>iteration()</i>	(-)	Minutu baten iragatea simulatzen duen metodoa. <i>Conduct</i> klaseko <i>refreshxandy()</i> metodoarekin pertsona guztien posizioak eguneratzen dira, eta distribuzioaren arabera <i>Illness</i> klaseko metodoekin kutsatze probabilitatea kalkulatu da toki esanguragarrietako pertsona osasuntsuentzat .
<i>testpeople()</i>	(-)	<i>Testinglist</i> listako pertsonak testatzen duen metodoa.
<i>DWorktime()</i>	(-)	Lanegunetan 8 lan ordu simulatzen dituen metodoa. Pertsona bakoitzaren jardura orduerdiro <i>Conduct</i> klaseko <i>getWTplace()</i> metodoa erabiliz zehaztuko da.
<i>DWhometime()</i>	(-)	Lanegunetan 8 ordu atsedean simulatzen dituen metodoa. Pertsona bakoitzaren jardura orduerdiro <i>Conduct</i> klaseko <i>getHTplace()</i> metodoa erabiliz zehaztuko da.

Metodoa	Argumentuak (Datu mota)	Azalpena
<code>DWleisuretime()</code>	(-)	Lanegunetan 8 ordu aisialdi simulatzen dituen metodoa. Pertsona bakoitzaren jarduera orduerdiro <i>Conduct</i> klaseko <code>getLTplace()</code> , <code>getGroupP11()</code> eta <code>getGroupP12()</code> metodoak erabiliz zehaztuko da.
<code>WEhometime()</code>	(-)	Jaiegunetan 12 ordu atsedean simulatzen dituen metodoa. Pertsona bakoitzaren jarduera orduerdiro <i>Conduct</i> klaseko <code>getHTplace()</code> metodoa erabiliz zehaztuko da.
<code>WEleisuretime()</code>	(-)	Jaiegunetan 12 ordu aisialdi simulatzen dituen metodoa. Pertsona bakoitzaren jarduera orduerdiro <i>Conduct</i> klaseko <code>getLTplace()</code> , <code>getGroupP11()</code> eta <code>getGroupP12()</code> metodoak erabiliz zehaztuko da.
<code>passingday()</code>	(-)	Egun baten iragatea simulatzen duen metodoa. Simulazioko pertsona guztiei <code>nextday()</code> metodoa aplikatzen zaie. <i>Totestlist</i> <i>testinglist</i> izatera pasatzen da eta epidemiari buruzko datuak <i>Data</i> klasean gordetzen dira.

Pertsonen atributuen esleipena

Metodo hauetaz gain, pertsonen atributuak era koherentean esleitu ahal izateko hainbat metodo eta atributu gehituko dira. Atal honetan hauetan sakondu gabe, era laburrean familia, kuadrilla eta lanpostu banaketa nola egin den azalduko da.

Etxe eta familiak esleitzeko `allocatefamilies()` metodoa eraikiko da. *Etxe* bakoitzean 2, 3 edo 4 pertsona biziko dira. 0 eta 60 urte bitarteko pertsonak 3 edo 4 pertsonako etxetan antolatuko dira, bi pertsona nagusi eta bat edo bi gazte edo hurrekin. Nagusien artean 0-3 urteko adin diferentzia ezarriko da, baita haurren edo gazteen artean. Gazte edo haur eta nagusien arteko adin diferentzia 30 urte ingurukoa izango da. 60 urtetik gorako pertsonak binakako etxetan biziko dira, adin paretsuko pertsona batekin batera.

Familia talde bat 2, 3 edo 4 familiek osatuko dute. Familia taldeak ausaz esleituko dira, kideen adinean erreparatu gabe.

Talde sozialak adinaren arabera esleituko dira `allocatesocialgroups()` metodoarekin. Belaunaldiko edo adin bakoitzeko hamar lagun talde ezberdin egongo dira, guztiak kide kopuru berdinekin. Taldeak ausaz banatuko dira adin bereko pertsonen artean.

Amaitzeko *lantokiak* ere adinaren arabera esleituko dira `allocateworkplaces()` metodoarekin. 0-23 urte bitarteko pertsoneri *ikasgela* bat esleituko zaie adinaren arabera. Belaunaldiko 6 ikasgela daudenez, ikasgela bakoitzak 20 ikasle izango ditu. 24-62 urte bitarteko pertsoneri lantoki bat esleituko zaie ausaz, beti ere **2.8 Taulako** datuak irizpide izanik eta sektoreka langile kopuru egokia esleituz.

2.2.4 *Data* klasea

Azkenik *Data* klasea eraikiko da. Klase honek epidemiaren garapena aztertzeke balioko diguten datuak gorde eta CSV fitxategi baten idazteke balioko du.

Atributuei dagokionez, balio osoak izango diren adierazleak izango ditugu. Esaterako, etxean ematen diren kasuak aztertu nahi izanez gero *homecases* eta *totalhomecases* atributuak eraikiko dira.

Metodoei dagokionez berriz, hauek manipulatzeko balio dituzten metodoak eraikiko dira, esaterako, `addhomecase()` edo `resethomecases()`.

Sortu beharreko atributu kopurua nahi den informazioaren arabera izango da. Kutsapenak eman diren tokiei, adin tarteei, ospitaleratu kopuruari eta beste hainbat adierazleri buruzko informazio zehatza nahi izanez gero, hauei dagozkien atributu eta metodoak gehitu beharko lirarteke.

Amaitzeko, pasatako azken 14 egunetako kasu detektatuen erregistroa gordez, inzidentzia tasa kalkulatzeko ahalbidetuko duen metodo bat ere gehituko da. Azken metodoa datu hauek guztiak bi CSV fitxategitan gordeko duen metodo bat eraikiko da. Lehen fitxategian eguneko datuen eboluzioa gordeko da, esaterako, egunero detektatutako kasu kopurua. Bigarrenetan berriz simulazio epe osoko datuak gordeko dira, sendatu eta hildako kopuru absolutuak esaterako.

3. Kapituluia

Epidemiaren hedapenaren simulazioa eta emaitzen analisia

Eraiki den metodoa probatzeko, *Illness* klasea erabiliz COVID-19 gaixotasunaren eredu bat eraikiko da, klasearen atributuak gaur egun ezagutzen diren datuetara egokituz. Programa ebaluatzeko lehenik Ekainetik eta Abendura bitarteko denbora tartea simulatuko da, eta emaitzak periodo horretako neurketekin alderatuko dira. Amaitzeko programaren hainbat aplikazio proban jarriko dira, aztarnatze sistema, distantziamendu eta maskaren erabileran oinarritutako neurri ezberdinak probatuz eta eragina aztertuz.

Gaixotasuna sortzen duen SARS-CoV-2 birusa kontaktu zuzen, zeharkako kontaktu, aire partikula eta *Flügge* tanten bidez transmititzen denaren ebidentziak aurkitu dira [15, 16, 17], bide nagusiak aire bidez ematen direnak izanik. Transmisio metodo hau α_i balioak doitu simulatuko da. Hasieran ausaz aukeratutako bost pertsona kutsatuko dira.

Asintomatiko, ospitaleratze eta heriotza ratioak adinaren arabera definituko dira, adin tarte bakoitzeko balioak behatutako ratioen arabera ezarriz. Maskaren eragina simulatzeko berriz, igorritako patogenoen ehuneko determinatu bat iragazten dutela onartuko da [18].

3.1 Pandemiaren eboluzioa: Ekaina-Abendua

Programa ebaluatzeko Ekainak 19tik Abenduak 12ra bitarteko denbora tartea simulatuko da, seihileko honetan eman ziren neurriak era kronologikoan errepikatuz. Lorturiko datuak Oñatin epe berean detektatu ziren kasuekin alderatuko dira. Eraikitako programan inplementatuko diren neurriak Euskal Autonomia Erkidegoan epe horretan ezarri ziren neurriak irizpide hartuz ezarriko dira:

Ekainak 19 - Abuztuak 18: Normaltasun berrira itzulia. Maskaren erabilera derri-gorrezkoa da. Eskolak itxita egongo dira. Kultura, kirol eta ostalaritza lokalek %60-ko edukiera izango dute, azken hauen kasuan terrazak irekita egongo dira.

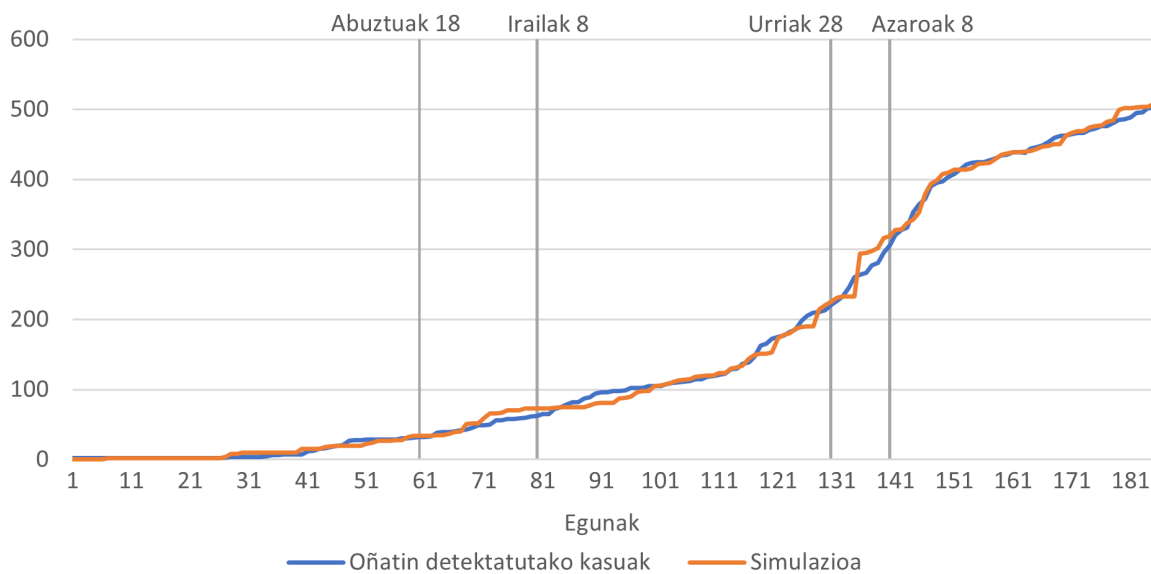
Abuztuak 19 - Irailak 7: Bilkurak 10 pertsonetara mugatzen dira. Ostalaritza lokalen itxiera 1:00etara aurreratzen da eta soilik eserita kontsumitu daiteke.

Irailak 8 - Urriak 27: Ikasturte hasiera eta eskolen irekiera.

Urriak 28 - Azaroak 7: Etxeratze agindua 23:00etan. Bilkurak 6 pertsonetara murrizten dira. Ostalaritza lokalen itxiera 21:00etara aurreratzen da.

Azaroak 8 - Abenduak 12: Etxeratze agindua 22:00etara aurreratzen da. Ostalaritza, kultura eta kirol zentroyen itxiera.

Herriko toki banaketa Oñatiko enplegu datuetan oinarritu da (**2.8 Taula**). Pertsona bakoitza aldiune bakoitzean zein tokitan egongo den berriz aisialdi denbora eta jardueri buruz eginiko inkesta soziologikoetan [13, 14] oinarrituz eraikitako portaera profilek determinatzen dute. Baldintza hauek programan portaera profil hauek moldatzen inplementatu dira, lorturiko emaitzak ondorengoak izanik:

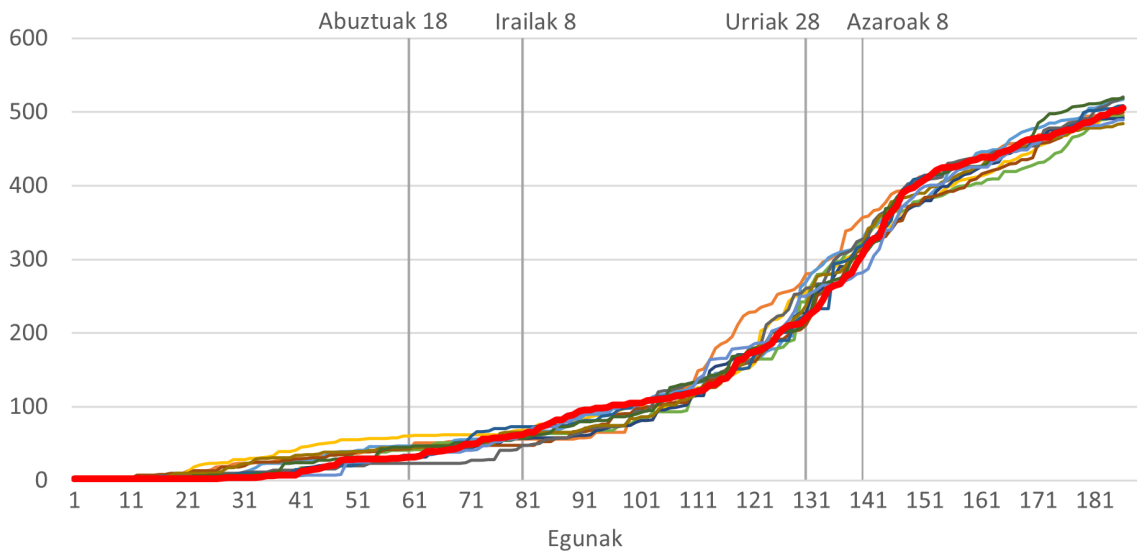


3.1. Irudia: Urdinez: Oñatin Ekainak 19-Abenduak 18 bitartean guztira detektatutako kasuen eboluzioa. **Laranja:** Simulazioan guztira detektatutako kasuak egunez egun.

Lehenik hedapen erritmoa eta aplikatu diren neurriek honetan izandako eragina aztertuko dira. Kontuan izan behar da gaixotasuna eta transmisio metodoa deskribatzeko erabili diren eredu matematikoak hurbilpenak direla, eta programak herri isolatu bat simulatu duela, inguruko herriekiko elkarrekintzak kontuan izan gabe. Hala ere, hurbilpenak tar-teko, behatutako kasu kopuru eta hedapenaren joera esponentziala era egokian simulatu dela konprobatu ahal izan da.

Beste aldetik portaera eta neurri aldaketen eragina aztertu behar da. Neurri ez-farmazeutikoak diseinatzea eta hauen eragina aztertzea ahalbidetzen duen metodoa eraikitzea denez helburu, ezinbestekoa da programak neurri hauei era egokian erantzutea. **3.1 Irudian** konprobatu daitekeen bezala, neurri aldaketak adierazten dituzten zuzen bertikal gris-oren ostean aldaketa bat dago gaixotasunaren hedapen erritmoan. Gaixotasunak Oñatin izandako hedapenaren erritmo aldaketak simulazioan emandakoekin alderatuz gero, programaren erantzuna egokia dela ikusi daiteke. Neurrien eragina berehalakoa ez dela eta 5-10 egunetara nabaritzen hasten direla ere konprobatu daiteke.

Diseinatutako programan ausazko metodoak erabiltzen direnez, posible da emaitzek bariantza bat izatea. Hurrengo pausua hainbat simulazio egitea izango da, alde batetik programaren bariantza aztertu ahal izateko, eta bestetik, informazio epidemiologiko fidagarriagoa izateko, detekzio tasa eta kutsatze eremuei buruzko datuak besteak beste.



3.2. Irudia: Hamar simulazioetan guztira detektaturiko kasuak Oñatin guztira detektatutako kasuen eboluzioarekin (zuzen **gorria**) alderatuz.

Kasu kopurua eta hedapen erritmo aldaketa emaitza guztietan berdintsuak direla konprobatu daiteke. Neurri ez-farmazeutikoen eraginen atzerapena ere ikusi daiteke. Hala ere, joera hauek simulazio guztietan errepikatzen diren arren, emaitzen artean bariantza nabari bat ageri da. Pertsona bakoitzak burutuko dituzten ekintza, posizioa eta bere osasun egoerak izango duen eboluzioa ausaz determinatzen direnez, bi simulazio eta emaitza berdina izatea ezinezkoa da.

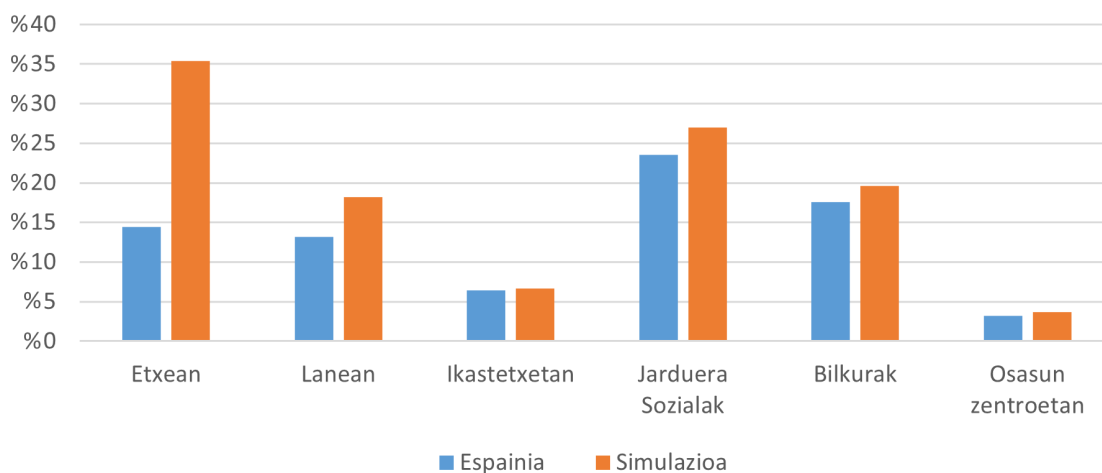
Bariantza honek ondorioak ateratzeko eragozpen bat dirudien arren, aintzat hartu behar da COVID-19 gaixotasunak eragindako pandemiak herriz-herri izandako hedapen erritmoa oso ezberdina izan dela, baita demografia eta enplegu banaketa berdintsuko herrietan ere. Hedapen erritmo eta bariantzari buruz gehiago sakonduko da hurrengo atalean.

Pertsonen portaera era egokian programatu dela ziurtatzeko, eremu ezberdinetan emandako kutsatze kopurua aztertu eta lorturiko datuak Espainian erregistratutako datuekin alderatuko dira. Ondorengo taulan Espainian eremu ezberdinetan emandako agerraldi eta kutsatu kopurua ikusi daitezke¹:

3.1. Taula: Espainian detektaturiko COVID-19 agerraldi eta kasu kopurua kutsapen eremu ezberdinetan [19, 20].

	Agerraldiak	Kasuak	%	Kasuak agerraldiko
Guztira	29027	246500	%100	8.6
Ikastetxetan	2545	15827	%6.42	6.2
Osasun zentroetan	769	7912	%3.21	10.3
Familia eremuak²	5659	35540	%14.42	6.3
Lantokiak	3397	32506	%13.19	9.7
Jarduera sozialak	8125	58102	%23.57	7.2
Lagun/familia bilkurak	(-)	(-)	%17.55	(-)
Mistoak	5093	45571	%18.49	8.9

Ondoren, simulazioko kutsatze eremu nagusietan emandako kutsatu kopurua eta Espainian esparru ezberdinetan emandako kutsatu kopurua alderatuko dira:



3.3. Irudia: **Urdinez:** Espainian COVID-19 agerraldietan detektatutako kasuak kutsatze eremuko. **Laranja:** Simulazioan eremu bakoitzean emandako kutsatzeak kutsatze osoekiko ehunekotan adieraziz.

¹CCAES-en irizpidearen arabera, 3 kutsatu edo gutxiago eman diren egoerak ez dira agerraldi gisa kontuan hartu, eta honelako egoeratan emandako kutsatuen datuak taula honetatik at daude.

²Etxebizitza bakarrean soilik eragin duten agerraldiak ez dira kontuan hartu eta bertan emandako kutsatuak taula honetatik at daude.

Konparaketa hau kualitatiboa da, eta ondorioak atera aurretik hainbat xehetasun azpimarratzea komeni da. Lehenik 11380 biztanleko herri bat lagin txikia dela kontuan izan behar da. Emaidza zehatzagoak lortu nahi izanez gero ezinbestekoa izango litzateke gutxienez eskualde edo probintzi bat simulatzea, enplegu banaketa eta egitura demografikoa Espainiako datuekin konparagarria izan daitezen.

Espainiako datuei dagokionez, aipatu behar da adierazitako datuak agerraldiei lotutako kasu detektatu kopuruak direla, hau da, agerraldiekin lotuta ez dauden kutsatuak ez direla kontuan hartu, etxe bakarrean edo 3 pertsonatik beherako taldeei eragin dietenak esaterako. Simulazioan berriz kutsatu guztiak hartu dira kontuan, baita detektatu gabe geratu diren kasu asintomatikoak ere.

Faktore hauen eragina arbuiagarria ez den arren, simulazioko eremu nagusietan emandako kutsatu kopuruaren arteko erlazioak Espainian detektatutakoen antzekoak direla ikusi daiteke, ezberdintasun nabariena etxean emandako kasuena izanik.

Amaitzeko, aipagarria da izandako detekzio tasa. Emaitzetan adierazitakoa kasu detektatuak izan dira, ez guztira kutsatuak. Simulazioetan ezarri den aztarnatze metodoa COVID-19 gaixotasunak eragindako pandemiaren aurkako protokoloan ezarritako aztarnatze metodoa imitatzeke helburuarekin ezarri da. Detektatu den kasu bakoitzeko pertsona detektatuaren familia talde, talde sozial eta lantokiko kideei, hau da, gertuko kontaktuei, testa burutu zaie. Aztarnatze metodoaren arabera detekzio tasa hau aldatu daiteke. Honetaz gain, detektatuaren etxeko kideak konfinatu dira etxeko kide guztiek testean emaitza negatiboa eman arte.

Aztarnatze metodo honekin lorturiko detekzio tasa %64.02 izan da, kutsatuen %35.98ak gaixotasuna asintomatikoki eta detektatu gabe igaro duelarik. Aztarnatze metodoen inplementazioak epidemietan gaixotasuna pasa duen gizartearen ehunekoaren estimazio bat egiteko balio dezake, eta hurrengo atalean ikusiko den bezala, metodo efektiboagoak diseinatzeke aukera ematen du.

3.2 Aztarnatze metodo eta hedapen abiadura

Aurreko atalean neurri ez-farmazeutikoek pandemiaren hedapenean duten eragina aztertu ahal izan da, baita eraiki den programak neurri hauek inplementatzeko eta era egokian simulatzeko duen gaitasuna.

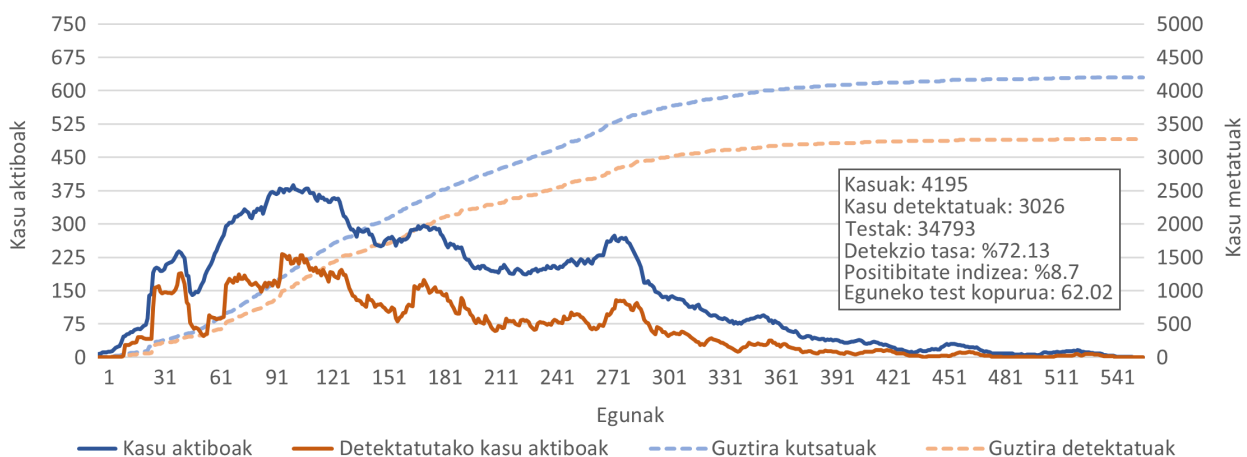
Aipagarriak izan diren bi fenomeno nagusi behatu dira. Lehen simulazioen arteko bariantza izan da. Bariantza hau COVID-19 gaixotasunak eragindako pandemian egitura demografiko eta enplegu banaketa antzekoak dituzten herrietan behatu den arren, arazo bat izan daiteke neurrien eraginak era zehatzean aztertu nahi badira. Bigarrena berriz detekzio tasa izan da. Erabilitako aztarnatze metodoarekin programan eman diren kasuen ia bi heren detektatu dira.

Faktore hauen kausa posibleak era sakonago batean aztertzeko, aztarnatze metodo ezberdinak diseinatuko dira eta hauek implementatuz hainbat simulazio burutuko dira. Simulazio hauetan guztira detektatutako kasuez gain, guztira kutsatutako pertsona kopurua eta uneoro simulazioan dauden kasu aktiboak irudikatuko dira.

Bestalde, aurreko ataleko simulazioa pandemia hedapenean eten da. Egoera honetan pre-sintomatikoak edo hurrengo egunetan detektatu daitezkeen kasuak egon daitezke, detektatu gabe geratu direnak. Kasu hauek detekzio tasan eragingo dute. Estimazio zehatzagoa egiteko simulazio epea luzatuko da kasu aktibo kopurua 0-ra murriztu arte. Horretaz gain, aztarnatze metodoaren eragina soilik aztertu nahi denez, simulazio osoan zehar aurreko atalean Irailak 8-Urriak 28 epean aplikatutako neurriak ezarriko dira.

3.2.1 1. Metodoa: Aztarnatze metodo estandarra

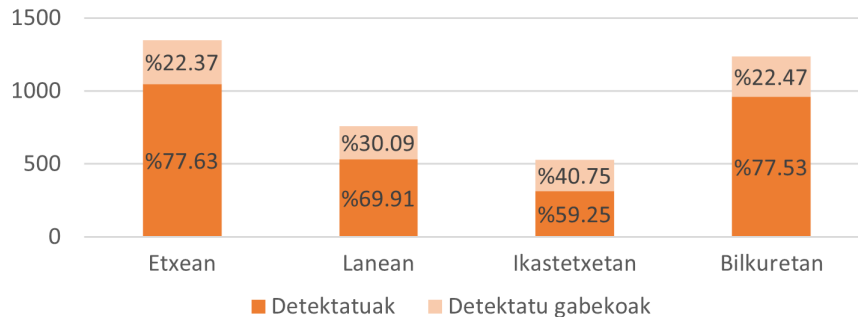
Erabiliko den lehen metodoa aurreko atalean erabilitako bera izango da, gertuko kontaktuak testatu eta etxeke kideak konfinatzen dituen. Baldintza hauetan eginiko simulazioan lorturiko kasuen eboluzioa, kasu aktiboak ezkerreko ardatzarekiko eta guztira metatutako kasuak berriz eskuineko ardatzarekiko adierazi direlarik, ondorengo izan da:



3.4. Irudia: Pandemiaren eboluzioa egunez egun pertsona detektatuen talde sozial, familia talde eta lantoki bereko kideak testatu eta etxe bereko kideak konfinatzen dituen aztarnatze metodoa ezarri.

3.4. Irudian ikusi daitezkeen hazkunde erritmoa **3.1 Irudian** eta **3.2. Irudian** ikusitakoarekin alderatuz, ezberdintasun nagusia hazkunde erritmoan ikusi daiteke. Lehen atalean eginiko simulazioetan hedapenak hazkunde esponenziala jarraitzen du, hala ere, simulazio denbora luzatuz gero, hazkunde erritmo hau potentzia-hazkunde bilakatzen dela konprobatu daiteke. Joera hau bat dator COVID-19 gaixotasunak sorturiko pandemian herrialde ezberdinetan behatutako joerarekin, non hasierako hazkunde erritmoa esponenziala den arren, denbora igaro ahala potentzia-hazkunde bilakatu den [21].

Kutsatze eremu nagusietan emandako kasu kopurua eta detekzio tasa aztertuz gero, emaitzak ondorengoak izan dira:



3.5. Irudia: Simulazioko eremu nagusietan emandako kutsatze kopurua, ehunekoek eremu bakoitzeko detekzio tasa adierazten dutelatrik.

3.5 Irudiko datuak 3.3 Irudiko datuekin alderatuz, ezberdintasun nabariena bilkuretan emandako kutsatu kopurua da. Bilkuretan emandako kutsatuen areagotzearen arrazoi nagusia bilkuretako pertsona kopuru maximoa 10 pertsonan ezarri izana da.

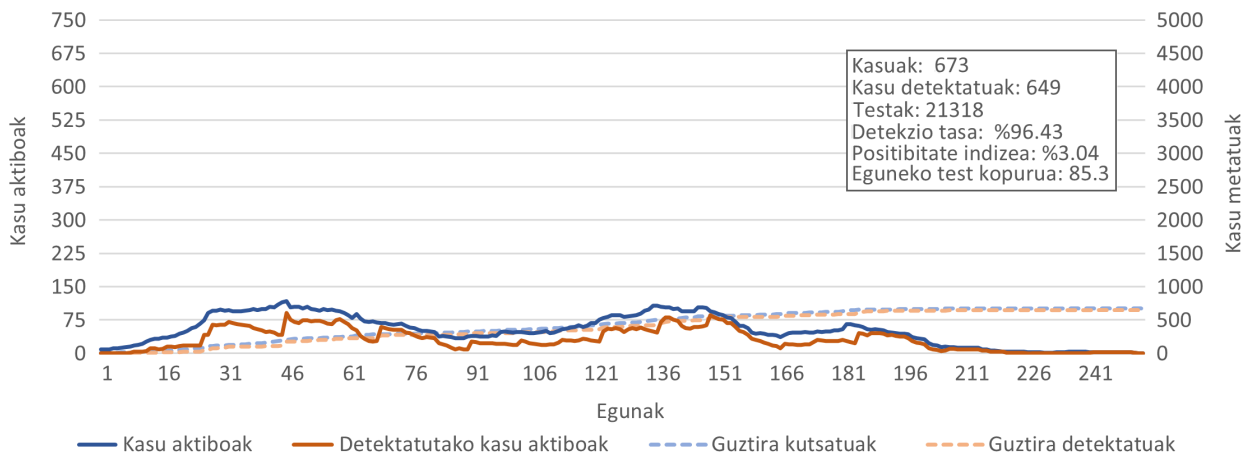
Amaitzeko, aipagarria da lantoki eta ikastetxetako emandako detekzio tasa baxuagoa. Definitu den aztarnatze metodoan, gertuko kontaktuak, hau da, testa burutuko dutenak, familia talde, lantoki edo ikasgela eta talde sozial bereko kideak dira. Kutsatze bat etxean edo bilkura baten ematen denean kutsatua beti gertuko kontaktu bat izango da. Ikastetxe eta lantokietan ordea, posible da zona komunetan gertuko kontaktuak ez diren pertsonen artean kutsatzeak ematea, eta aztarnatze metodoa dela eta, kasu hauek zailagoak dira detektatzeko. Hau dela eta, ikastetxe eta lantokiek detekzio tasa baxuagoak dute.

3.2.2 2. Metodoa: Gertuko kontaktuen zabaltzea

Hurrengo simulazioan aztarnatze metodoa aldatuko da detekzio tasa baxuena duten eremuetan testatze masiboak eginez. Aztarnatze metodo honetan detektatutako kasu bakoitzean lantoki edo ikasgela bereko pertsonak soilik testatu beharrean, eremu berdineko pertsona guztiak testatuko dira. Esaterako, eskola baten ikasle batek positibo emanez gero, ikasgela bereko kideak soilik testatu beharrean irakaskuntza zentroko ikasle eta irakasle guztiei testa burutuko zaie.

Lorturiko emaitzak (ikus **3.6. Irudia**) aztertuz gero eta hazkundera **3.4 Irudian** ikusitakoarekin alderatu, gertuko kontaktuen taldea zabalduz kasu kopurua eta epidemiaren iraupena nabarmen murriztu dela konprobatu daiteke. Alde batetik kasu kopuru osoa 4195 kasutik 673 kasutara murriztu da, %83.96 hain zuzen. Bestalde, epidemiaren iraupena %54.37 murriztea lortu da, 561 egunetik 257 egunera.

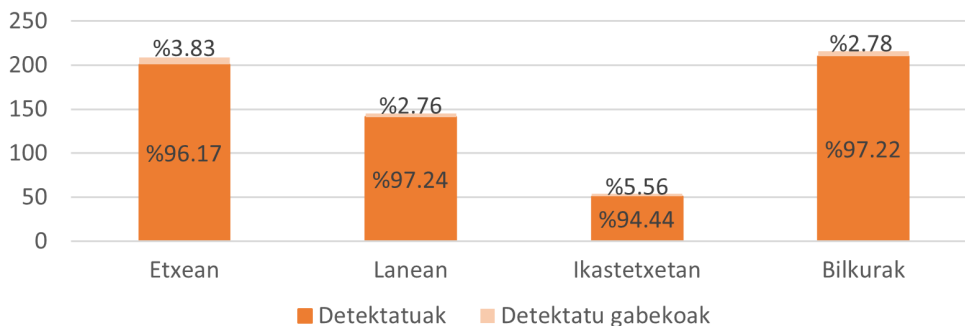
Amaitzeko test kopurua aztertu behar da. Eguneko burututako test kopurua handiagoa izan arren, epidemiaren iraupena eta kasu kopurua nabarmen murriztu direnez, orotara test gutxiago burutu dira. Aipagarria da bi simulazioetan eguneko eginiko test kopuruen arteko diferentzia nabarmena izan ez arren, 64.4 eta 85.3 test eguneko hurrenez hurren, kasu detektatuko eginiko test kopurua nabarmen handiagoa dela bigarren metodoarekin. Lehen simulazioan 11.5 test burutu dira kasu detektatuko, bigarreanean berriz 32.8.



3.6. Irudia: Pandemiaren eboluzioa egunez egun pertsona detektatuen talde sozial, familia talde eta lan eremu bereko kideak testatu eta etxe bereko kideak konfinatzen dituen aztarnatze metodoa ezarritz.

Kasu detektatuko test kopuru altuaren eragina detektatutako kasu aktiboetan ikusi daiteke. Detektatutako kasu aktiboen zuzena **3.4 Irudian** adierazitakoarekin alderatuz gero, test masiboagoetan oinarritutako metodoarekin detekzio maximo nabariagoak izan direla konprobatu daiteke, egun horietan test kopuru handiak burutu diren seinale.

Kutsatze eremu nagusietan emandako kasu kopurua eta detekzio tasa aztertzen badira, emaitzak ondorengoak izan dira:



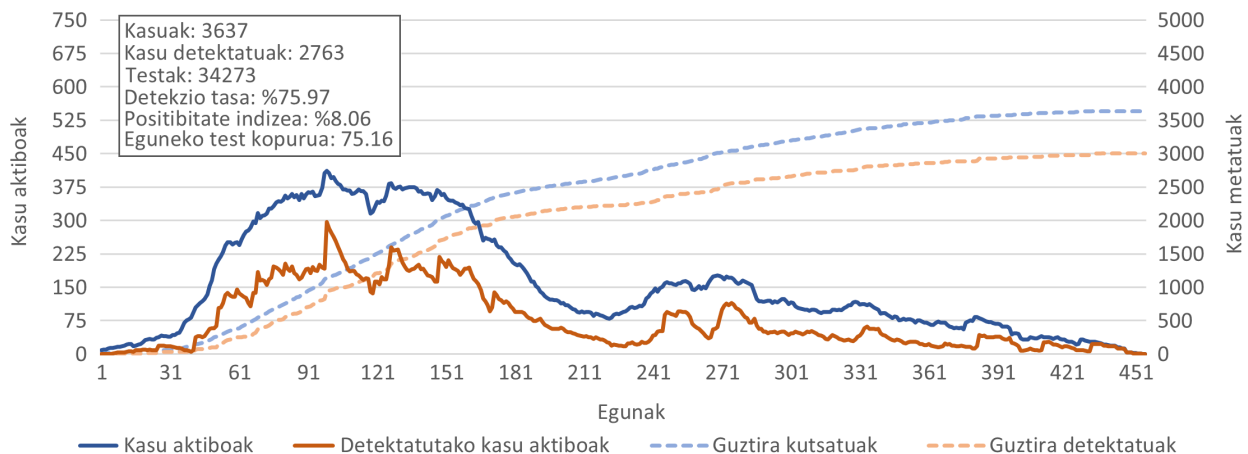
3.7. Irudia: Simulazioko eremu nagusietan emandako kutsatze kopurua, ehunekoek eremu bakoitzeko detekzio tasa adierazten dutelatrik.

Testatze metodoa lan eremuetan soilik zabaldu den arren, **3.7. Irudiko** emaitzak aztertuz, implementatutako metodoak lan eremutik at emandako kutsatuak detektatzeko ere baliagarria dela konprobatu daiteke. Alde batetik lan eremuetan eginiko test kopuru handiagoaren ondorioz bertako detekzio tasa inkrementatu da. Bestetik, lantokietan detektatutako kasu asintomatikoek etxe edo bilkuretan kutsatutako pertsonen aztarna jarraitzeko balio dute, toki hauetan emandako agerraldiak detektatzen lagunduz.

3.2.3 3. Metodoa: Ausazko testak

Kontuan izan behar da testak baliabide mugatua direla eta kutsatuen aztarnatze eta testatze lanek esfortzu eta langile beharra dutela. Testak burutzeko azpiegitura egokia eta langile nahikoa izan ezean, kontaktu zirkulua zabaltzen oinarritzen diren estrategiak, aurreko simulazioan implementatu dena esaterako, ez dira bideragarriak.

Arazo hau konpontzeko, hurrengo simulazioan erdibideko aztarnatze metodo bat inplementatuko da. Lehen simulazioko aztarnatze metodo bera erabiliko da, hau da, kasu positibo bat detektatzen den bakoitzean pertsona detektatuaren talde sozial, familia talde eta lantoki bereko kideak testatu, eta etxebizitza bereko kideak konfinatuko dira. Hala ere, asteen behin lantoki bakoitzean detektatu gabeko pertsona bat ausaz testatuko da. Metodo honekin lorturiko epidemiaren hedapena ondorengoa izan da:

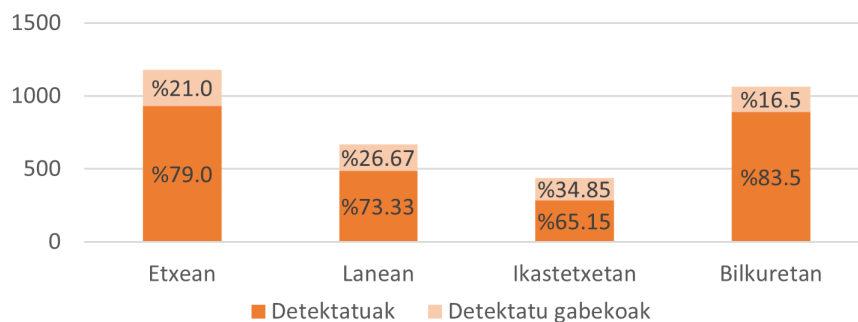


3.8. Irudia: Pandemiaren eboluzioa pertsona detektatuen talde sozial, familia talde eta lantoki bereko kideak testatu eta etxe bereko kideak konfinatzen dituen aztarnatze metodoa ezarriz eta lantokietan ausazko testatzeak gehituz.

Adierazle epidemiologikoak aztertuz gero eta erabili den lehen aztarnatze metodoarekin lortutakoekin alderatu, ausazko testatzeak gehituz kutsatu kopurua %13.3 murriztu dela konprobatu daiteke, 4195 kasutik 3637 kasura. Bestalde epidemiaren iraupena %18.72 murriztu da, 561 egunetik 456 egunera.

Eginiko test kopurua aurreko metodoekin lorturikoekin alderatuz, eguneko eginiko test kopuruak tarteko balio bat hartzen duela ikusi daiteke, 75.16 test eguneko hain zuzen. Test kopuru osoa eta kasu detektatuko eginiko test kopurua ordea lehen metodoarekin lortutako balioen berdintsuak dira, 34173 test eta 12.4 test kasu detektatuko hurrenez hurren. Detektatutako kasu aktiboen zuzena 2. metodoarekin lortutakoarekin alderatuz, orain lorturikoa leunagoa dela ere ikusi daiteke, eguneko eginiko test kopuruak bariantza txikiagoa izan duen seinale.

Kutsatze eremu nagusietan emandako kasu kopurua eta detekzio tasa aztertuz gero, emaitzak ondorengoak izan dira:



3.9. Irudia: Simulazioko eremu nagusietan emandako kutsatze kopurua, ehunekoek eremu bakoitzeko detekzio tasa adierazten dutelarik.

Kutsatze eremu nagusietan izandako kutsatze eta detekzio tasak alderatzen badira, ausazko testak gehituz eremu guztietan emandako detekzio tasaren inkrementua homogeneoa izan dela konprobatu daiteke.

3.2.4 Hedapen erritmoa

Aztarnatze metodoak eta hauek dituzten eraginak aztertzen amaitzeko, diseinatutako hiru metodoekin epidemiak izandako hedapen abiadurak alderatu eta hauetan eragin duten faktoreak zeintzuk izan diren eztabaidatuko da.

Eredu epidemiologiko sinpleenetan hedapen erritmoak kutsatu kopuruarekiko eta immunizatu gabeko biztanleria kopuruarekiko du menpekotasun nagusia, pertsona kutsatuak direlako gaixotasuna transmititu dezaketenak eta osasuntsuak kutsatu daitezkeenak hain zuzen. Hala ere, eraikitako programarekin eginiko simulazioetan erlazioa konplexuagoa dela konprobatu daiteke.

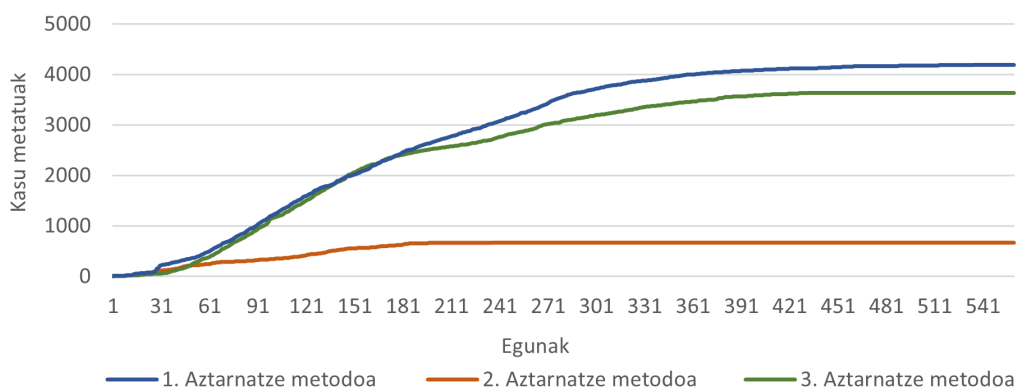
Diseinatu den metodoan detektatutako kasu aktiboak konfinatu direnez, transmisio eragile nagusiak detektatu gabeko kasu aktiboak dira. **3.4. Irudiko, 3.6. Irudiko eta 3.8. Irudiko** emaitzak aztertuz, eta detektatu gabeko kasu aktiboak (kasu aktiboen eta detektatutako kasu aktiboen arteko diferentzia) eta guztira kutsatuak aztertzen badira, menpekotasun hori aztertu daiteke.

Kasu aktiboen eta detektatutako kasu aktiboen arteko diferentzia txikia den denbora tartetan guztirako kasuen zuzenak hazkunde motelagoa duela konprobatu daiteke, bi zuzenen arteko diferentzia handia denean berriz, alderantzizkoa gertatzen da.

Orokorrean detektatu gabeko kasu aktiboen eta kasu aktiboen artean erlazio zuzen bat egongo da, batez ere biztanleria lagin handietan. Hala ere erlazio hau aztarnatze metodoekin aldatu daitekeela konprobatu da, detektatu gabeko kasu kopurua murriztuz eta epidemiaren hedakuntza erritmoa nabarmen motelduz.

Aipagarria da ere hiru simulazioetan kasu aktiboen oszilakortasun arin bat ikusi dela. Detektatutako kasu aktiboak asko jaisten badira kasu aktibo kopurua nulua ez delarik, kutsatzeek gorakada nabariak eman ditzakete, olatu erako portaerak emanik.

Amaitzeko eta aztarnatze metodoak era zehatzagoan alderatzeko hiru simulazioetako kutsatu kopuru osoak gainezarriko dira:



3.10. Irudia: Urdinez: Lehen aztarnatze metodoarekin izandako guztirako kasu kopurua. **Berdez:** Bigarren aztarnatze metodoarekin izandako guztirako kasu kopurua. **Laranjaz:** Hirugarren aztarnatze metodoarekin izandako guztirako kasu kopurua.

Emaitzak gainbegiratzuz aurreko ataletan ondorioztatu dena azpimarratu daiteke. Gertuko kontaktuak kontsideratzen diren taldea zabalduz diseinatu den metodoena izan da efektiboena. Ausaz eginiko testak ere neurri txikiago baten kasu kopurua murrizteko gai dira, hala ere metodo honen onurak denboran aurrerago nabaritzen hasten dira, biztanleriaren zati adierazgarri batek gaixotasuna pasatu ostean. Honen arrazoi nagusia ausazko testak gaixotasuna pasatu ez dutenen pertsoneri burutu zaizkiela izan da.

Hala ere, kontuan izan behar da atal honetan diseinatu diren aztarnatze metodoak ez direla aplikagarriak izateko helburuarekin diseinatu. COVID-19 pandemian aplikatu den aztarnatze metodoa espezifikoagoa izan da, pertsona batek azken egunetan izandako kontaktuen erregistroetan oinarritu baita. Aztarnatze metodo zehatzagoak diseinatu nahi izanez gero posible izango litzateke simulazioan pertsona bakoitzaren kontaktu erregistroa gordetzea edo kribatu masiboak implementatzea.

Azkenik, ezin da gainbegiratu simulazio hauetan herrien artean eman daitekeen kutsatuen berrelikadura alde batera utzi dela. Herritik kanpo ikasi edo lan egiten duten pertsonak eta herrian ikasi edo lan egiten duten kanpoko pertsonak rol nagusia izan dezakete hedapenean, eta posible da errealitatean gaixotasunaren hedapen abiadura lan honetan lortutakoa baino handiagoa izatea. Dena dela, atal honen helburua aztarnatze estrategia ezberdinen eragina aztertzea izan da, eta emaitzak aztertuz aztarnatze metodoen eragina eta testatze metodo egokien garrantzia azpimarratu ahal izan da.

3.2.5 Emaitzen bariantza

3.10. Irudira itzuliz gero, simulazioak aurrera egin ahala kutsatze erritmoak perturbazio nabaririk jasan ez duen arren, lehen egunetan simulazio ezberdinen artean **3.2. Irudian** behatutako bariantza bera eman dela konprobatu ahal izan da.

Aurretik aipatu den bezala, hedapen abiaduran eragiten duen faktore garrantzitsuenetako bat detektatu gabeko kasu aktibo kopurua da. Detektatu gabeko kasu aktibo gutxi daudenean, simulazio hasieran esaterako, ausaz ematen diren kutsatzeak edo pertsona bat sintomatiko bilakatzeak eragin handiagoa dute, proportzioan balio hauek nabarmen aldatzen baitira.

Hau garbiago ikusi daiteke gela baten pertsona kutsatu eta pertsona osasuntsu bana dauden egoera bat definitzen bada. Kutsatzeko probabilitatea %50 baldin bada, hainbat simulazio burutuz gero pertsona osasuntsua simulazioen erdian kutsatu egingo da, eta gainontzekoetan ez, estatistikoki bariantza handia izanik. Egoera aldatuz eta 100 gela definitu, non bakoitzean pertsona kutsatu eta osasuntsu bana dauden, eta kutsatzeko probabilitate berriro ere %50 baldin bada, simulazio gehienetan pertsonen erdia kutsatuko da gutxi gorabehera, estatistikoki bariantza nabarmen murriztuz.

Sistema kaotikoekin analogia egin daiteke. Detektatu gabeko pertsona gutxi dauden uneak puntu kritiko edo ezegonkor gisa kontsideratu daitezke, eta bertan emandako perturbazioek (kutsatze edo kasu bat sintomatiko bilakatzeak) sistemaren eboluzioa determinatzen dute eta luzera eragin nabariagoa dute detektatu gabeko kutsatu asko dauden unetan emandako kutsatze edo kasu detektatuek baino.

Portaera honek enplegu banaketa eta egitura demografiko berdintsuak dituzten tokietan COVID-19 pandemiak izandako hedapen abiadura ezberdinak ere azaldu dezake, baita hainbat herrik pandemiaren olatu ezberdinetan izandako erritmoak ezberdinak izatea. Egitura demografiko eta enplegu banaketa antzekoak dituzten herriak alderatzen badira pandemiak bertan izan duen hazkunde erritmoa nabarmen aldatu daitekeela konprobatu da. Eskala handitzen bada eta esaterako berdintsuak diren probintzietan emandako hedapen abiadurak alderatu ordea, bariantza txikiagoa izan ohi da.

Hau garrantzitsua da diseinatu den metodoa eremu txikietan erabili nahi bada. Lan eremu bat simulatu nahi izanez gero esaterako, kontuan izan behar da emaitzek bariantza bat izan ditzaketela eta zuhurrena ondorioak atera aurretik hainbat simulazio egitea dela.

3.3 Segurtasun metodo pertsonalen eragina

Orain arte diseinatu den metodoa neurri ez-farmazeutikoen eragina simulatzeko eta aztarnatze metodo ezberdinen implementazioa aurrera eramateko gai dela konprobatu da.

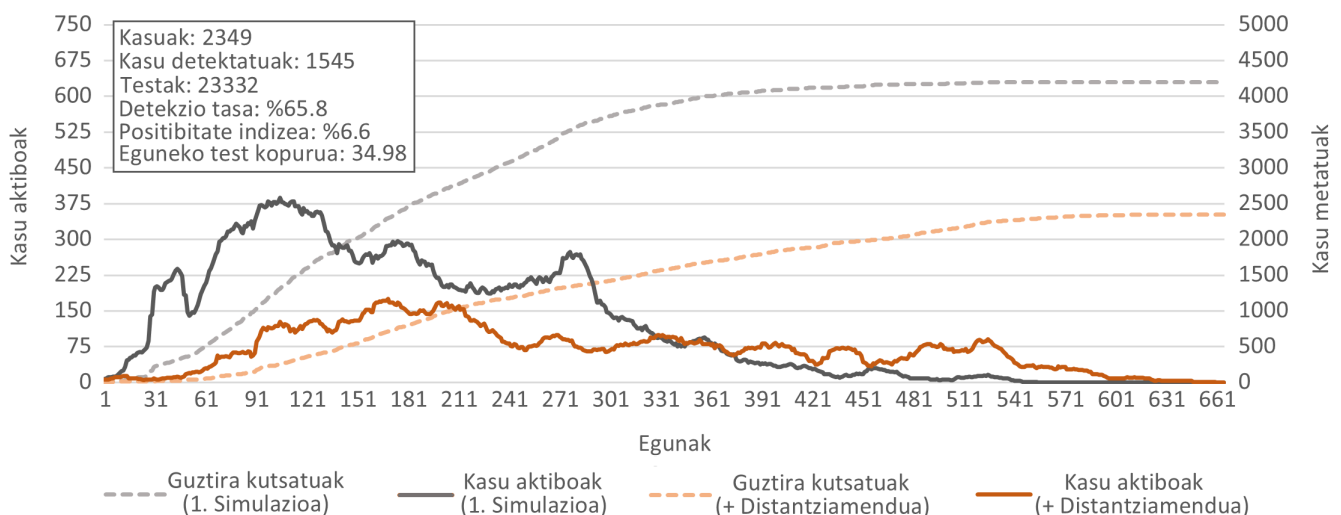
Amaitzeko, azken atal honetan eraikitako programan segurtasun edo prebentzio neurri pertsonalak inplementatuko dira, maskaren erabilera eta distantziamendu soziala hain zuzen. Simulazioan inplementatuko diren baldintza eta aztarnatze metodoa **3.2.1 Atalekoak** izango dira eta emaitzak bertan lorturikoekin alderatuko dira.

3.3.1 Distantziamendu soziala

Pertsonen arteko distantzia handituz, (1.6) ekuazioan ikusi daitekeen bezala, esposizioa murriztu daiteke, kontaktuan jarritako *Flügge* tanta, aire partikula eta kontaktu zuzeneko transmisio metodoekin erlazionatutako patogeno dentsitatea distantzia handituz murrizten baita.

Aipagarria da esparru sozial batzutan distantziamendua ez dela bideragarria, osasun arazoak direla eta behar bereziak dituzten pertsonen edo osasun zentroen kasuan adibidez. Hala ere, simulazio honetan distantziamendu sozialak izan ditzaken ondorio epidemiologikoak aztertu eta ulertzea izango da.

Atal honetan **3.2.1 Atalean** erabilitako programari aldaketa bat egingo zaio, pertsonen arteko distantzia 0.5 m inkrementatuz. Lorturiko emaitzak ondorengoak izan dira:



3.11. Irudia: Grisez: 3.4. Irudiko simulazioan emandako kasu aktiboak eta guztira kutsatuak. **Laranja:** Distantziamendu soziala handituz lorturiko kasu aktibo eta guztirako kutsatuak.

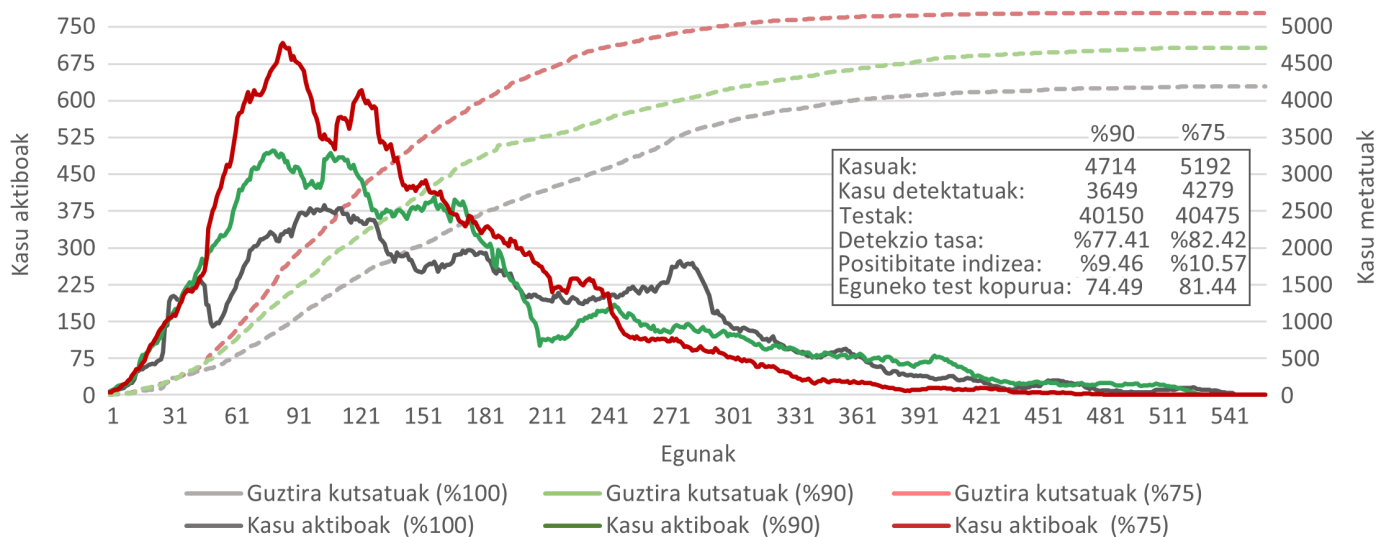
Eragin aipagarriena kasuen murrizketa da. Bestalde, kutsatze ratioa ere murriztu egin da, eta kasu aktiboak alderatzen badira kutsatzeak denboran era homogeneousagoan eman direla konprobatu daiteke, kurba epidemiologikoa leunduz. Epidemiaren iraupena luzatu den arren, kasu aktibo kopurua txikiagoa izatea komenigarri da kasu ospitaleratuek osasun zerbitzuen kapazitate maximoa gainditu ez dezaten.

Aipagarria da ere detekzio tasan izandako murrizketa. Kausa posible bat eginiko test kopuru txikiagoa izan daiteke. Beste arrazoi bat, hurrengo atalean sakonago aztertuko den bezala, transmisio bide ezberdinek gertuko kontaktuak kutsatzeko duten garrantzia izan daiteke.

3.3.2 Maskaren erabilera

Maskaren erabilerak aire bidez ematen diren *Flügge* tanta eta aire partikulen hedapena murrizteko balio dute. Eginiko simulazio guztietan maskaren erabilera tasa %100 izan da etxetik kanpo eman diren jardueretan, erreferentzi gisa izandako epean zehar maskararen erabilera legez nahitaezkoa izan baita. Etxeko jardueretan berriz, soilik pertsona detektatuek erabili dituzte maskarak.

Atal honetan etxetik at emandako jardueren maskaren erabilera murriztuko da eta %90 eta %75eko erabilera tasekin simulazio bana burutuko da. Emaitzak alderatuz maskaren erabilerak dituen ondorio epidemiologikoak ulertu eta interpretatzea izango da helburu. Simulazioen lorturiko emaitzak ondorengoak izan dira:



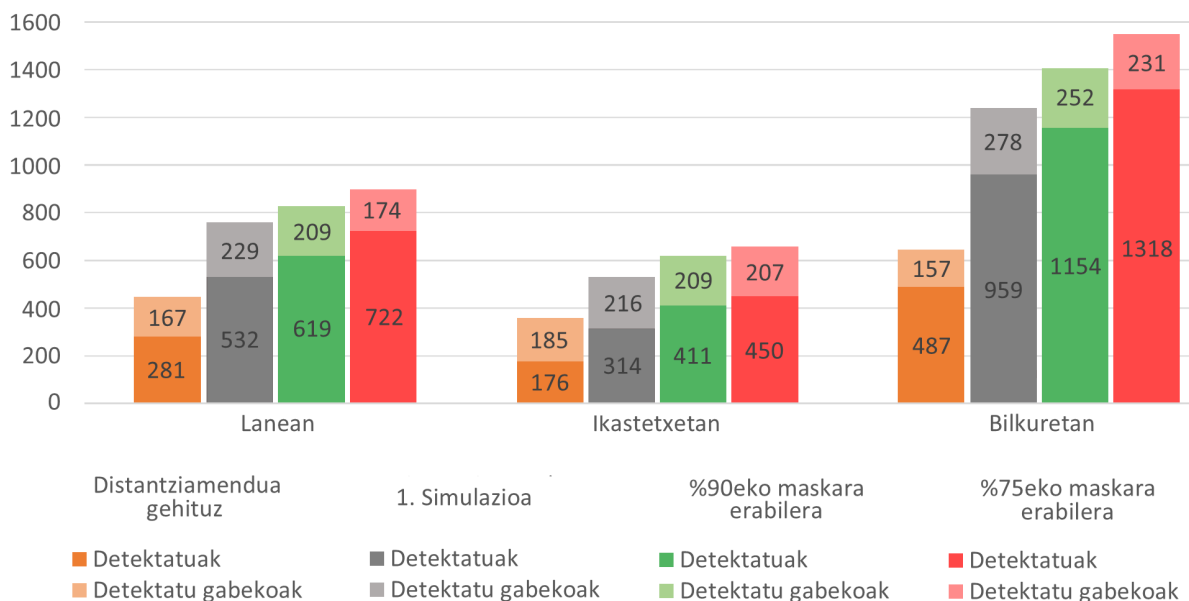
3.12. Irudia: Simulazioen emaitzak maskaren erabilera tasa ezberdinentzat. **Grisez:** **3.4. Irudiko** simulazioan emandako kasu aktiboak eta guztira kutsatuak. **Berdez:** Maskaren erabilera tasa %90 izanik lorturiko kasu aktibo eta guztirako kutsatuak. **Gorritz:** Maskaren erabilera tasa %75 izanik lorturiko kasu aktibo eta guztirako kutsatuak.

Maskaren erabilera murriztu ahala behatu daitekeen lehen ondorioa kasuen inkrementua izan da. Distantziamendu sozialak bezala, maskaren erabilerak kutsatze ratioa murrizten lagundu du. Erabilera jaitsi ahala kutsatze erritmoa handitzen dela konprobatu daiteke hedapena bizkortuz, eta ondorioz, epidemiaren iraupena laburtuz.

Kutsatze kopuruaren inkrementuaren eta epidemiaren iraupenaren laburpenaren ondorioz, maskaren erabilera baxuena izan duten simulazioetan kasu aktibo kopuru handiagoa izan da. Maskaren erabilera murriztu ahala kurba epidemiologikoak maximo nabariagoa duela ikusi daiteke, kutsatu gehienak denbora tarte labur batean metatuz.

Eginiko test kopuruari eta positibitate indizeari erreparatuz eta distantziamendu sozialak izandako eraginekin alderatuz, ondorio antzekoak atera daitezke. Maskaren erabilera inkrementatu ahala kasu kopurua murriztu eta epidemiaren iraupena luzatu da, eguneko egin beharreko test kopurua murriztuz. Hala ere, kasu kopurua jaitsi ahala, positibitate indizea ere murriztu egin dela konprobatu da.

Positibitate indizearen aldaketa, aurreko atalean bezala, test kopuruaren aldaketaren ondorio izan daiteke. Hala ere, portaera hau hobeto ulertzeko aztertu diren lau simulazioetan eremu ezberdinetan emandako kutsatze eta detektatu kopuruak aztertuko dira:



3.13. Irudia: Eremu nagusian emandako kutsatzeak, zutabe ilunek eta bertako balioek kasu detektatuak eta zutabe argiek eta bertako balioek detektatu gabeko kasuak adierazten dutelarik. **Grisez:** 3.4. **Irudiko** simulazioko emaitzak. **Laranja:** Distantziamendu soziala handituz lorturiko emaitzak **Berdez:** Maskaren erabilera tasa %90 izanik lorturiko emaitzak. **Gorritz:** Maskaren erabilera tasa %75 izanik lorturiko emaitzak.

Emaitzak aztertuz, distantziamendu eta maskaren erabilerak kasu detektatu kopuruan eragin nabariagoa duela ikusi daiteke, eta kasu ez detektatuen kopurua nabarmen gutxiago aldatzen dela.

Definitutako metodoan bi pertsonen arteko distantzia txikia denean transmisio bide nagusiak kontaktu zuzen, *Flügge* tanten eta aire partikulen bidezkoak dira. Hurbiltasun eta esposizio denbora dela eta, gertuko kontaktu gisa definitu diren talde sozial, familia talde eta lantoki edo ikasgela bereko kideak dira transmisio metodo nagusi hauen bitartez gehien kutsatutakoak. Gertuko kontaktuak ez diren pertsonen kasuan berriz, distantziarekiko menpekotasunik ez duen zeharkako kontaktu bidez emandako transmisio metodoak garrantzi handiagoa du.

Hau dela eta, kontaktu zuzen, *Flügge* tanten eta aire partikulen bidezko transmisioan eraginak duten distantziamendu eta maskaren erabilerak efektiboagoak dira gertuko kontaktuen kutsatzeak murrizten, eta ez dute eraginkortasun bera gertuko kontaktuak ez diren pertsonen kutsatzeak murrizteko. Gertuko kontaktuen detekzioa errazagoa denez, distantziamendu eta maskaren erabileraren zeharkako ondorio bat detekzio tasa murriztea da. Hau da, detekzio tasa ez da detektatzen zailagoak diren kasu gehiago daudelako igo, detektatzen errazak diren kasu gutxiago daudelako baizik.

4. Kapituluia

Ondorioak

Lan honen helburua nagusia gizarteko lagin espezifikoaren analisi epidemiologikoak burutzeko ahalbidetuko duen metodo bat diseinatzea izan da. Hau lortzeko gizartea ikuspuntu makroskopikotik aztertzen duten eredu epidemiologiko klasikoetatik aldendu eta gizabakoetan oinarritutako metodo bat hutsetik diseinatu da.

Metodo hau inplementatu ahal izateko transmisio metodo sinplifikatu baten oinarri matematikoa garatu da, eta *Java* programazio-lengoaian oinarrituz pertsonei, tokiei eta gaixotasunei dagozkien *software* ereduak eraiki dira.

Metodoaren bideragarritasuna probatzeko, Oñati herriko eta enplegu datu eta banaketa demografikoan oinarrituta herri bat eraiki da. Azkenik, COVID-19 gaixotasunari buruz izandako ezagutza oinarri gisa eta pandemiaren hedapen datuak erreferentzi gisa hartuta, pandemiak herrian izandako eraginak simulatu dira.

Lan honek bi ildo nagusi izan ditu. Alde batetik, lanaren helburu nagusiarekin erlazionaturik, analisi epidemiologikoak burutzeko baliagarri den metodoaren diseinua eta honen ebaluazioa. Bestetik, diseinatutako metodoa inplementatuz eginiko azterketa espezifikoari dagokion atala. Egitura honi jarraituz, ondorioak ere bi zatitan banatuko dira, lehendabizi burutu diren Oñati herriaren simulazioen emaitzen analisisia eginik, eta amaitzeko, diseinatu den metodoa era orokorrean aztertuz, honen aplikagarritasun, abantaila eta desabantailak era sakonagoan eztabaidatuz.

Lortutako emaitzen analisisekin hasi baino lehen, eraikitako programan simulazioetan eragin duten sinplifikazioak aipatu behar dira. Eraiki den herria herri isolatu gisa eraiki da, hau da, ez dago inolako elkarrekintzarik inguruko herriekin, eta honek epe luzera eragin nabariak izan ditzake epidemiaren garapenean. Bestalde, gaixotasuna deskribatzeko erabili den eredu matematikoan hainbat sinplifikazio egin dira.

Hala ere, sinplifikazioak tarteko, lortu diren emaitzak COVID-19 gaixotasunak eragindako pandemian erregistratu diren datuekin bat datoztela konprobatu ahal izan da, hedapen erritmo, kutsatze esparru eta neurrien eraginei dagokionez batez ere.

Bestalde, maskaren erabilera eta distantziamendu sozialen eragina aztertu eta ulertu ahal izan da, eta prebentzio metodo pertsonal hauek kurba epidemiologikoa leuntzeko duten garrantzia azpimarratzea lortu da. Testatze metodoek duten garrantzia ere frogatu ahal izan da, hauek pandemiaren hedapenean duten eragina aztertuz.

Hasiera batean intuitiboak ez diren fenomenoak ere aztertu ahal izan dira, eredu epidemiologiko klasikoetan erduztatzen zailagoak direnak. Aipagarrienak detekzio tasa eta talde immunitatea dira. Aztarnatze metodoak inplementatuz detekzio tasa onduztatu ahal izan da, bide batez gaixotasuna igaro duen gizartearen ehunekoa estimatzeko aukera emanez. Talde immunitatea ere aztertu ahal izan da, eredu epidemiologiko klasikoetan talde immunitatea gizarte osoak pasa duenean lortu ohi den arren, eginiko simulazioetan ez da halakorik behatu. Prebentzio metodo pertsonalek detekzio tasan duten eragina ere aztertu ahal izan da, hauen garrantzia azpimarratuz.

Talde immunitate eta detekzio tasa hauek neurri ez-farmazeutiko eta testatzeak azaldu ditzakete, hala ere, gogoan izan behar da eraiki den programan hainbat hurbilpen eta sinplifikazio egin direla, eta azken ondorioak atera baino lehen komenigarria litzateke programaren hainbat xehetasun hobetzea, aire bidezko transmisio eredu sofistikatuagoak diren eredu matematikoetan oinarrituz (jariakinen mekanikan oinarritutakoak adibidez) edo gaixotasun eta erantzun immunologikoa era zehatzagoan deskribatzen dituzten klaseak inplementatuz.

Bestalde kontuan izan behar da simulazioetan olatu epidemiologiko bakarra soilik aztertu dela. Eredu zabalago bat simulatuz gero, inguruko herriak kontuan hartuz, posible da herrien arteko kutsatze berrelikadurak olatu bat baino gehiago sortzea, talde immunitatea beranduago lortuz.

Amaitzeko, diseinatu den metodoa ebaluatuko da, ez soilik simulatu den eremuan zentratuz, baizik eta gizabanakoetan oinarrituta diseinatu den eredu epidemiologikoak eskaintzen dituen aukerak aztertuz. Eraiki den programa intuitiboak diren klase sinplez oinarritu da, pertsonak eta tokiak esaterako, eta hor datza diseinatutako metodoaren abantaila nagusiak, doiketa eta egokitzapenak era sinplean egitea ahalbidetzen baitu.

Alde batetik, programaren malgutasunak egoera ezberdinetara egokitzeko gaitasuna eskaintzen du. *Illness* klasea eraldatuz programa gaixotasun edo andui ezberdinetara moldatu daiteke, edo transmisio metodo konplexuago bat inplementatu. *Population*, *Town* eta *Conduct* klaseak eraldatuz portaera soziologiko ezberdineko biztanleria eta herrialde-tara ere moldatu daiteke, epidemien hedapena toki ezberdinetan aztertu eta konparatu ahal izateko.

Bestetik, dimentsio ezberdinetako eremuak simulatzeko gaitasuna dela eta helburu ezberdineko programak eraikitzea ahalbidetzen du.

Esaterako, transmisio metodoa doitzuz eremu txikietako kutsatze probabilitatea zehaztasun handiz simulatu daiteke. Garapen honek ikasgela, bulego edota fakultate baten pertsona distribuzio seguruak bilatzen laguntzeko balio dezake.

Beste aplikazio bat eskala handiagoko eremuak simulatzea izan daiteke. Oñatiren kasuan egin den bezala, metodo honetan oinarrituz eskualde edo hiriak simulatu ditzaketen programak eraiki daitezke. Programa hauek prebentzio eta segurtasun neurri efektiboak diseinatzeko balio ditzakete. Konprobatu ahal izan den bezala, aztarnatze metodo eraginkorragoak diseinatzeko edo balio dezake. *Person* klasean *txertatua* osasun egoera gehituz, txertatze estrategiak diseinatzeko ere baliagarria izan daiteke.

Amaitzeko, eskala handiagoan baliagarriagoak diren metodo eredu epidemiologiko konpartimentalak doitzeko ere lagungarria izan daiteke. Eredu konpartimentaletan zaila da neurri ez-farmazeutiko batek edo efektu klimatologiko batek nola eragin dezaken auresatea. Eraiki den metodoan ordea faktore hauek *Illness* eta *Conduct* klasean erraz inplementatu daitezke, honela R_0 oinarritzko birsorkuntza tasan izan dezaken eragina auresanez eta eskala handiagoak simulatzeko erabilitako eredu epidemiologikoak doitzuz.

Hau dena dela eta, lan honekin pandemia egoera larrietan prebentzio eta segurtasun neurri eraginkorragoak diseinatzeko baliagarria den metodo bat garatu dela esan liteke.

Bibliografia

- [1] World Health Organization. “Pneumonia of unknown cause - China”
<https://www.who.int/csr/don/05-january-2020-pneumonia-of-unknown-cause-china/en/> - 2020ko Martxoa.
- [2] Li Y, Liang M, Yin X, Liu X, Hao M, Hu Z, Wang Y, Jin L.
“2020 COVID-19 Epidemic Outside China: 34 founders and exponential growth.”
medRxiv, 2020.
- [3] World Health Organization.
<https://covid19.who.int/> - 2020ko Abendua.
- [4] Ferguson, N. et al. “Report 9: Impact of Non-Pharmaceutical Interventions (NPIs) to Reduce COVID19 Mortality and Healthcare Demand.” *Imperial College London*, 2020.
- [5] “Medidas de contención y desescalada adoptadas por los gobiernos europeos en la primera ola del coronavirus (enero-julio 2020)” *Barcelona Centre for International Affairs*, 2020.
- [6] Eubank, S., Guclu, H., Anil Kumar, V. et al. “Modelling disease outbreaks in realistic urban social networks”. *Nature* 429, 180-184, 2004.
- [7] Kermack, W. O. McKendrick, A. G. “A contribution to the mathematical theory of epidemics”. *Proc. R. Soc. Lond.*, 1927.
- [8] Wells, W. F. “On air-borne infection. Study II. Droplets and droplet nuclei”. *American Journal of Hygiene*, 1934.
- [9] Feng, Y., Marchal, T., Sperry, T., Yi, H. “Influence of wind and relative humidity on the social distancing effectiveness to prevent COVID-19 airborne transmission: A numerical study”. *Journal of Aerosol Science*, 2020.
- [10] K.W. Mui, L.T. Wong, C.L. Wu, Alvin C.K. Lai, ”Numerical modeling of exhaled droplet nuclei dispersion and mixing in indoor environments”. *Journal of Hazardous Materials*, 2009.
- [11] Grassly, N. C., Fraser, C. “Mathematical models of infectious disease transmission”. *Nature Reviews Microbiology*, 2008.

- [12] Tillett, R., Sevinsky, J., Hartley, P., et al. “Genomic evidence for a case of reinfection with SARS-CoV-2”. *bioRxiv*, 202
- [13] “Informe de igualdad 2019 > Ocio y vida social”. *Eustat*, 2019.
- [14] “Estudio N° 3123 | Encuesta social general Española (ESGE)”. *CIS*, 2015.
- [15] “Modes of transmission of virus causing COVID-19: implications for IPC precaution recommendations”. *WHO*, 2020.
- [16] Ong, S. W. X. et al, “Air, surface environmental, and personal protective equipment contamination by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) from a symptomatic patient”, *J. Am. Med. Assoc.*, 2020.
- [17] Cai, J., Sun, W., Huang, J., Gamber, M., Wu, J., He, G., “Indirect Virus Transmission in Cluster of COVID-19 Cases, Wenzhou, China, 2020”. *Emerging Infectious Diseases*, 2020.
- [18] Mittal, R., Meneveau, C., Wu, W., “A mathematical framework for estimating risk of airborne transmission of COVID-19 with application to face mask use and social distancing”. *Physics of Fluids*, 2020.
- [19] “Actualización n° 265. Enfermedad por el coronavirus (COVID-19). 22.01.2021”. *Ministerio de Sanidad*, 2020.
- [20] “Actualización n° 296. Enfermedad por el coronavirus (COVID-19). 22.01.2021”. *Ministerio de Sanidad*, 2021.
- [21] Komarova, N. L., Schang, L. M., Wodarz, D., “Patterns of the COVID-19 pandemic spread around the world: exponential versus power laws”, *Journal of The Royal Society Interface*, 2020.

A Eranskina

Programaren dokumentazioa

Class Hierarchy

Classes

- `java.lang.Object`
 - `Epidemic.Conduct` (in [A.1](#), page 55)
 - `Epidemic.Illness` (in [A.2](#), page 62)
 - `Epidemic.Peoplelist` (in [A.3](#), page 67)
 - `Epidemic.Person` (in [A.4](#), page 69)
 - `Epidemic.Place` (in [A.5](#), page 75)
 - `Epidemic.Population` (in [A.6](#), page 77)
 - `Epidemic.Town` (in [A.7](#), page 87)

Package Epidemic

Package Contents

Page

Classes

Conduct	55
Illness	62
Peoplelist	67
Person	69
Place	75
Population	77
Town	87

A.1 Class Conduct

Class containing methods that emulate different behaviour profiles during leisure time. Each conduct profile determines the probability for any given person to do different activities during leisure time. Profiles are divided in nine subprofiles, which are applied depending on the age of the given person.

The possible behavioural profiles are the following:

Standard: All places are open and accessible, the probabilities are defined based on surveys.

Closedhostelry: Bars and restaurants are closed and inaccessible. All other places are open and accesible.

Confined: All places except shops are closed and inaccessible.

A.1.1 Declaration

```
public class Conduct
    extends java.lang.Object
```

A.1.2 Constructor summary

[Conduct\(\)](#)

A.1.3 Method summary

[allocatefamilygroups\(\)](#)
[allocatehostelry1\(\)](#)
[allocateplacesweekend\(\)](#)
[allocateplacesweektime\(\)](#)
[allocatepublicplaces1\(\)](#)
[allocatepublicplaces2\(\)](#)
[allocateweekendfamilygroups\(\)](#)
[allocateweekendsocialgroups\(\)](#)
[getagegroup\(int\)](#)
[gethostelryplace\(\)](#)
[getnursinghome\(\)](#)
[getplaceDWfamilyactivity\(Person\)](#)
[getplaceDWsocialactivity\(Person\)](#)
[getplaceIA\(Person\)](#)
[getplaceWEfamilyactivity\(Person\)](#)
[getplaceWESocialactivity\(Person\)](#)

```
getpublicplace1(int)
getpublicplace2(int)
getpublicplace3()
getpublicplace4()
getshoppingplace()
getstreet()
setgroupactivities(boolean)
setprofile(String)
shuffle(int[])
```

A.1.4 Constructors

- **Conduct**

```
public Conduct ()
```

– Description

Class that determines the behaviour of the population during leisure time using behavioural profiles. Each profile is divided in nine subprofiles. Each subprofile indicates the probabilities for a person to do certain activity. Each subprofile consists of an array with 8 double values, each value indicating the probability for:

- 1: Probability to go to a bar, pub or cafeteria.
- 2: Probability to go to a shop.
- 3: Probability to go to a public place dedicated to enfants (e.g. parks)
- 4: Probability to go to a public place frequented by young people.
- 5: Probability to go to public places like the cinema, a museum, the libraries... (equally frequented by all adults)
- 6: Probability to go to public places frequented by elder people.
- 7: Probability to go to the nursing home.
- 8: Probability to be in the street.

If the randomized method fails to give a value which falls on the previously mentioned ranges, the place allocated will be one's own home.

A.1.5 Methods

- **allocatefamilygroups**

```
public void allocatefamilygroups ()
```

– Description

Method that allocates family groups to hostelry places during week.

- **allocatehostelry1**

```
public void allocatehostelry1()
```

- **Description**

- Method that allocates social groups to hostelry places during week.

- **allocateplacesweekend**

```
public void allocateplacesweekend()
```

- **Description**

- Method that allocates social and family groups to certain activities during weekends.

- **allocateplacesweektime**

```
public void allocateplacesweektime()
```

- **Description**

- Method that allocates social and family groups to certain activities during week time.

- **allocatepublicplaces1**

```
public void allocatepublicplaces1()
```

- **Description**

- Method that allocates social groups of children to public places.

- **allocatepublicplaces2**

```
public void allocatepublicplaces2()
```

- **Description**

- Method that allocates social groups of young people to public places.

- **allocateweekendfamilygroups**

```
public void allocateweekendfamilygroups()
```

– **Description**

Method that allocates family groups to hostelry places during week.

• **allocateweekendsocialgroups**

```
public void allocateweekendsocialgroups()
```

– **Description**

Method that allocates social groupsof children to public places.

• **getagegroup**

```
public java.lang.String getagegroup(int age)
```

– **Description**

Method returning the age group of a given age.

– **Parameters**

* **age** – Age of the person being evaluated.

– **Returns** – Age group of the person being evaluated.

• **gethostelryplace**

```
public int [] gethostelryplace()
```

– **Description**

Method that returns a random hosterly place, e.g. bar or cafeteria.

– **Returns** – Location of a random hostelry place in Town clas.

• **getnursinghome**

```
public int [] getnursinghome()
```

– **Description**

Method that returns the nursing home location.

– **Returns** – Location of the nursing home in the Town class.

• **getplaceDWfamilyactivity**

```
public int [] getplaceDWfamilyactivity(Person person1)
```

- **Description**

Method that, given a person, returns the place where it will be while doing a certain family activity.

- **Parameters**

- * `person1` – Person whose destination is needed.

- **Returns** – `Int[]` indicating the destination on the `Town` class.

- **getplaceDWsocialactivity**

```
public int [] getplaceDWsocialactivity(Person person1)
```

- **Description**

Method that, given a person, returns the place where it will be while doing a certain group activity. Several social groups of people between 8 and 17 years are allocated places for extracurricular activities. Social groups of people between 18 and 25 years are allocated places for extracurricular activities, sports, or public places. Social groups of people between 26 and 65 years are allocated hostelry places such as bars or cafeterias.

- **Parameters**

- * `person1` – Person whose destination is needed.

- **Returns** – `Int[]` indicating the destination on the `Town` class.

- **getplaceIA**

```
public int [] getplaceIA(Person person1)
```

- **Description**

Method that, given a person, returns the place where it will be. The place is selected randomly taking into account the current behaviour profile.

- **Parameters**

- * `person1` – Person whose destination is needed.

- **Returns** – `Int[]` indicating the destination on the `Town` class.

- **getplaceWEfamilyactivity**

```
public int [] getplaceWEfamilyactivity(Person person1)
```

- **Description**

Method that, given a person, returns the place where it will be while doing a certain family activity during weekends.

- **Parameters**

- * `person1` – Person whose destination is needed.

- **Returns** – `int[]` indicating the destination on the `Town` class.

- **getplaceWESocialactivity**

```
public int [] getplaceWESocialactivity(Person person1)
```

- **Description**

Method that, given a person, returns the place where it will be while doing a certain group activity on weekends. Several social groups of people between 8 and 17 years are allocated public places. Social groups of people between 18 and 93 years are allocated places hostelry places such as bars, restaurants or cafeterias.

- **Parameters**

- * `person1` – Person whose destination is needed.

- **Returns** – `int[]` indicating the destination on the `Town` class.

- **getpublicplace1**

```
public int [] getpublicplace1(int age)
```

- **Description**

Method that returns a pseudorandomly selected park or public place frequented by enfants. The place is not completely random, as age is a factor regarding which places can be frequented.

- **Parameters**

- * `age` – Age of the person being assigned to a public park.

- **Returns** – Location of the park in the `Town` class.

- **getpublicplace2**

```
public int [] getpublicplace2(int age)
```

- **Description**

Method that returns a pseudorandomly selected public place frequented by young people. The place is not completely random, as age is a factor regarding which places can be frequented.

- **Parameters**

- * `age` – Age of the person who is being assigned to a public place.

- **Returns** – place Location of the place frequented by young people in the Town class.

- **getpublicplace3**

```
public int [] getpublicplace3 ()
```

- **Description**

Method that returns a random public place like libraries, cinemas...

- **Returns** – Location of the public place in Town class.

- **getpublicplace4**

```
public int [] getpublicplace4 ()
```

- **Description**

Method that returns a random public place frequented by elder people.

- **Returns** – Location of the public place in the Town class.

- **getshoppingplace**

```
public int [] getshoppingplace ()
```

- **Description**

Method that returns a random shop.

- **Returns** – Random shop location on Town class.

- **getstreet**

```
public int [] getstreet ()
```

- **Description**

Method that returns open street location.

- **Returns** – Location for open street in the town class.

- **setgroupactivities**

```
public void setgroupactivities (boolean arepermitted)
```

- **Description**

Method that sets the ability to do group activities, if group activities are permitted.

- **Parameters**

- * `arepermitted` – Boolean indicating if group activities are permitted or not.

- **setprofile**

```
public void setprofile(java.lang.String profile)
```

- **Description**

Method that sets the ongoing behaviour profile.

- **Parameters**

- * `profile` – Profile of behaviour which is being set.

- **shuffle**

```
public int [] shuffle(int [] array)
```

- **Description**

Method use to shuffle an array randomly.

- **Parameters**

- * `array` – Array which is wanted to be shuffled.

- **Returns** – Shuffled array.

A.2 Class Illness

Properties of the illness. This class collects the parameters and the equations used to determine the evolution of the disease, and the probabilities each person has to advance to the next stage of the illness.

Parameters:

asymptomaticprob: Age depending probability of being asymptomatic.

hospitalprob: Age depending probability of a symptomatic person to develop severe symptoms and being hospitalized.

deathprob: Age depending probability of a hospitalized person dying.

α_1 : Age depending parameter used to adjust the probability of infection via respiratory droplets.

α_2 : Age depending parameter used to adjust the probability of infection via airborne transmission.

α_3 : Age depending parameter used to adjust the probability of infection via indirect contact.

α_4 : Age depending parameter used to adjust the probability of infection via direct contact.

r_0, d_0 : Parameters (double) used to adjust the range of infection probability.

maskfactor: Double indicating how much the usage of a mask reduces the risk of getting infected.

A.2.1 Declaration

```
public class Illness
    extends java.lang.Object
```

A.2.2 Constructor summary

[Illness\(\)](#)

A.2.3 Method summary

[asymptomaticprob\(Person\)](#)
[deathprob\(Person\)](#)
[getalpha1\(Person\)](#)
[getalpha2\(Person\)](#)
[getalpha3\(Person\)](#)
[getalpha4\(Person\)](#)
[getpathogendensity\(Person, Person, double, int\)](#)
[getprobability\(double\)](#)
[hospitalprob\(Person\)](#)
[nextstep\(Person\)](#)

A.2.4 Constructors

- **Illness**

```
public Illness ()
```

– Description

In this constructor the basic data structures are built. The parameters listed in the class description are inserted in dictionaries.

A.2.5 Methods

- **asymptomaticprob**

```
public double asymptomaticprob(Person person1)
```

- **Description**

Method used to determine the probability for a person to become asymptomatic.

- **Parameters**

- * **person1** – The person whose probability will be calculated.

- **Returns** – *asymptomaticprob* - The probability to be asymptomatic.

- **deathprob**

```
public double deathprob(Person person1)
```

- **Description**

Method used to determine the probability for a hospitalized person to die.

- **Parameters**

- * **person1** – The person whose probability will be calculated.

- **Returns** – *deathprob* - The probability of dying.

- **getalpha1**

```
public double getalpha1(Person person1)
```

- **Description**

Method used to determine the value of α_1 parameter for a given person.

- **Parameters**

- * **person1** – The person whose α_1 value is required.

- **Returns** – α_1 - The corresponding value of α_1 parameter.

- **getalpha2**

```
public double getalpha2(Person person1)
```

- **Description**

Method used to determine the value of α_2 parameter for a given person.

- **Parameters**

- * **person1** – The person whose α_2 value is required.
- **Returns** – α_2 - The corresponding value of α_2 parameter.

- **getalpha3**

```
public double getalpha3(Person person1)
```

- **Description**
Method used to determine the value of α_3 parameter for a given person.
- **Parameters**
 - * **person1** – The person whose α_3 value is required.
- **Returns** – α_3 - The corresponding value of α_3 parameter.

- **getalpha4**

```
public double getalpha4(Person person1)
```

- **Description**
Method used to determine the value of α_4 parameter for a given person.
- **Parameters**
 - * **person1** – The person whose α_4 value is required.
- **Returns** – α_4 - The corresponding value of α_4 parameter.

- **getpathogendensity**

```
public double getpathogendensity(Person person1, Person  
    person2, double area, int riskfactor)
```

- **Description**
Method used to determine the pathogen density a healthy person is being exposed to.
- **Parameters**
 - * **person1** – The healthy person whose exposure level is being calculated.
 - * **person2** – The infected person located in the same place of the healthy one.
 - * **area** – Double indicating the area of the place.
 - * **riskfactor** – Integer used to indicate the typo of place in which the pathogen is spreading.
- **Returns** – *pathogendensity* - The pathogen density the healthy person is exposed to.

- **getprobability**

```
public double getprobability(double pathogendensity)
```

- **Description**

Method that calculates the infection probability based on the pathogen density exposure. The probability is calculated using a hiperbolic tangent, therefore, for small values the transmission probability is directly proportional to the exposure level, and for higher values the probability converges towards its maximum value.

- **Parameters**

- * *pathogendensity* – Pathogen density the healthy person has been exposed to

- **Returns** – *probability* Probability of infection

- **hospitalprob**

```
public double hospitalprob(Person person1)
```

- **Description**

Method used to determine the probability for a symptomatic person to be hospitalized.

- **Parameters**

- * *person1* – The person whose probability will be calculated.

- **Returns** – *hospitalprob* - The probability to be hospitalized.

- **nextstep**

```
public void nextstep(Person person1)
```

- **Description**

Method used to advance an infected person to the next stage of the illness. Stages which don't happen always, such as hospitalization or asymptomatic are calculated according to their respective probabilities.

- **Parameters**

- * *person1* – Infected person who's going to advance to the next stage.

A.3 Class Peoplelist

Class which represents a list of people, composed of a $n \times 2$ resizable matrix filled with integers. Each row represents a person, where the two values of each row correspond to the indexes of the person in the People matrix in the Population class.

The main purpose of this class is to assign lists of people to places, contact groups of detected cases and more, to optimize other methods such as "illness tracking methods" by avoiding the need for sweeping the entire Population class.

A.3.1 Declaration

```
public class Peoplelist
    extends java.lang.Object
```

A.3.2 Constructor summary

[Peoplelist\(\)](#)

A.3.3 Method summary

[addperson\(int\[\]\)](#)
[alreadyin\(int\[\]\)](#)
[emptylist\(\)](#)
[getpeople\(\)](#)
[removeperson\(int\[\]\)](#)

A.3.4 Constructors

- [Peoplelist](#)

```
public Peoplelist ()
```

A.3.5 Methods

- [addperson](#)

```
public void addperson(int [] identity)
```

– **Description**

Method to add a person to a list by expanding the people matrix by adding the indexes to the last row. If a person is already in the list this method won't do anything.

– **Parameters**

- * **identity** – Couple of integers indicating the position of the added person in the Population matrix in the Town class.

• **alreadyin**

```
public boolean alreadyin(int [] identity)
```

– **Description**

Method that checks if a person is already in a list.

– **Parameters**

- * **identity** – Id of the person being checked.

– **Returns** – Boolean which indicates if the person is in the list.

• **emptylist**

```
public void emptylist()
```

– **Description**

Method which empties the list of people.

• **getpeople**

```
public int [][] getpeople()
```

– **Description**

Method that returns the list of people as a matrix.

– **Returns** – n x 2 Sized matrix of integers.

• **removeperson**

```
public void removeperson(int [] identity)
```

– **Description**

Method to remove a person from a list by reducing the People matrix by removing the row corresponding to the person being removed.

– **Parameters**

- * **identity** – Couple of integers indicating the position of the removed person in the People matrix in the Population class.

A.4 Class Person

Basic individual of the simulation. Each person has the following attributes:

- age**: Integer number indicating its age.
- countdown**: Integer indicating the number of days remaining for a person to evolve into the next health state.
- detected**: Boolean indicating if a person is known to have contracted the illness.
- masked**: Boolean indicating if a person is wearing a mask.
- workplace**: Array of integers (size = 3) identifying the place where it works.
- home**: Array of integers (size = 3) identifying the place where it lives.
- currentplace**: Array of integers (size = 3) indicating a person's location.
- socialgroup**: Integer identifying which social group it belongs to.
- familygroup**: Integer identifying which family group it belongs to. [1]
- xpos**: Physichal position on X axis.
- ypos**: Physichal position on Y axis.
- state**: String indicating the persons health state, which can be:
 - 0: uninfected
 - 1: latency
 - 2: incubation
 - 3: asymptomatic
 - 4: symptomatic
 - 5: hospitalized
 - 6: recovered
 - 7: dead

Not to confuse family and family group. Family: Group of 2-3-4 people sharing a home. Family group: A group consisting of people from 4-5 families, aka relatives (cousins, grandparents...).

A.4.1 Declaration

```
public class Person
  extends java.lang.Object
```

A.4.2 Constructor summary

[Person\(\)](#)

A.4.3 Method summary

[confine\(\)](#)
[deconfine\(\)](#)

`iscontagious()`
`nextday()`
`refreshposition(double)`
`setage(int)`
`setasymptomatic(int)`
`setconfined(boolean)`
`setdead()`
`setfamilygroup(int)`
`sethnumber(int)`
`sethome(int)`
`sethospitalized(int)`
`setincubating(int)`
`setinfected(int)`
`setmasked(Booleam)`
`setrecovered()`
`setsocialgroup(int)`
`setstate(int)`
`setsymptomatic(int)`
`setworkplace(int[])`
`test()`

A.4.4 Constructors

- **Person**

```
public Person ()
```

- **Description**

- Unit of the population. The only attributes set by default are not being masked and the uninfected health state.

A.4.5 Methods

- **confine**

```
public void confine ()
```

- **Description**

- Method used to confine a person.

- **deconfine**

```
public void deconfine ()
```

- **Description**

Method used to deconfine a person

- **iscontagious**

```
public boolean iscontagious ()
```

- **Description**

Method used to determine if a person is contagious based on its health state.

- **Returns** – Boolean indicating if a person is contagious.

- **nextday**

```
public void nextday ()
```

- **Description**

Method used to simulate a passing day on the illnesses development by subtracting a unit from the countdown value.

- **refreshposition**

```
public void refreshposition (double d)
```

- **Description**

Method used to refresh a person's physical position.

- **Parameters**

- * **d** – Double indicating the size of the place where the person is.

- **setage**

```
public void setage (int n)
```

- **Description**

Method used to set a persons age.

- **Parameters**

- * **n** – Integer indicating the new health state.

- **setasymptomatic**

```
public void setasymptomatic (int asymptomaticperiod)
```

– **Description**

Method used to transfer a person from a incubation state to a asymptomatic state.

– **Parameters**

* `asymptomaticperiod` – Days the disease stays in asymptomatic state.

• **setconfined**

```
public void setconfined(boolean conination)
```

– **Description**

Method used to set the conination state of a person.

– **Parameters**

* `conination` – Boolean indicating if the person is confined or not.

• **setdead**

```
public void setdead()
```

– **Description**

Method used to set a person in a dead state.

• **setfamilygroup**

```
public void setfamilygroup(int n)
```

– **Description**

Method used to assign a family to a person. People of the same family group will tend to hang up together or be on the same places during leisure time.

– **Parameters**

* `n` – Integer indicating the family group.

• **sethnumber**

```
public void sethnumber(int h)
```

– **Description**

Method used to convert a non mask wearing person into a mask wearing person.

– **Parameters**

* `h` – Integer indicating the corresponding hospital room number.

- **sethome**

```
public void sethome(int n)
```

- **Description**

- Method used to assign a home to a person. People of the same family will be on the same home during resting time.

- **Parameters**

- * `n` – Array of integers indicating the working place. Size = 3.

- **sethospitalized**

```
public void sethospitalized(int hospitalizedperiod)
```

- **Description**

- Method used to transfer a person from a symptomatic state to a hospitalized state.

- **Parameters**

- * `hospitalizedperiod` – Days the disease stays at the hospital.

- **setincubating**

```
public void setincubating(int incubationperiod)
```

- **Description**

- Method used to transfer a person from a latent state to a incubation state.

- **Parameters**

- * `incubationperiod` – Days the disease stays in incubation state.

- **setinfected**

```
public void setinfected(int latencyperiod)
```

- **Description**

- Method used to infect a person.

- **Parameters**

- * `latencyperiod` – Days the disease stays in latent state.

- **setmasked**

public void setmasked(java.lang.Boolean ismasked)

– **Description**

Method used to convert a non mask wearing person into a mask wearing person.

– **Parameters**

* **ismasked** – Boolean indicating if the person is masked.

• **setrecovered**

public void setrecovered()

– **Description**

Method used to set a person in a recovered state.

• **setsocialgroup**

public void setsocialgroup(int n)

– **Description**

Method used to assign a social group to a person. People of the same social group will tend to hang up together or be on the same places during leisure time.

– **Parameters**

* **n** – Integer indicating the social group.

• **setstate**

public void setstate(int n)

– **Description**

Method used to establish or modify a persons health state. If a person develops symptoms, it will automatically be regarded as a "detected" case.

– **Parameters**

* **n** – Integer indicating the age.

• **setsymptomatic**

public void setsymptomatic(int symptomaticperiod)

- **Description**

Method used to transfer a person from a incubation state to a symptomatic state.

- **Parameters**

- * `symptomaticperiod` – Days the disease stays in symptomatic state.

- **setworkplace**

```
public void setworkplace(int [] n)
```

- **Description**

Method used to assign a workplace to a person. People assigned the same workplace will be on the same place during work time.

- **Parameters**

- * `n` – Array of integers indicating the working place. Size = 3.

- **test**

```
public void test()
```

- **Description**

Method used to test a person. If a person is asymptomatic or has the illness in a latent state, the case will be detected.

This method can be further developed, introducing a random parameter that represents the rate of false positives or negatives.

A.5 Class Place

Basic physical location. Each place has a size, a PeopleList indicating the current people that are in it, and a risk level that helps further expanding the contamination model. The attributes of each place will be:

people: PeopleList indicating the people that currently are in each place.

size: Double indicating the width or length of a place. It should be noted that all places will be square, and Size here regards to an unidimensional value.

A.5.1 Declaration

```
public class Place
    extends java.lang.Object
```

A.5.2 Constructor summary

`Place()`

A.5.3 Method summary

`addperson(int[])`
`emptyplace()`
`getpeople()`
`removeperson(int[])`
`setsize(double)`

A.5.4 Constructors

- **Place**

```
public Place()
```

- **Description**

- Unit of a physical place. It will consist of a PeopleList which indicates the people placed inside.

A.5.5 Methods

- **addperson**

```
public void addperson(int [] identity)
```

- **Description**

- Method to add a person to a place by expanding the People matrix by adding position of the person being added on the population matrix on the Town class.

- **Parameters**

- * `identity` – Couple of integers indicating the position of the added person in the Population matrix in the Town class.

- **emptyplace**

```
public void emptyplace()
```

- **Description**

- Method to add a person to a place by expanding the People matrix by adding position of the person being added on the population matrix on the Town class.

- **getpeople**

```
public int [][] getpeople()
```

- **Description**

Method that returns the list of people in a certain place.

- **Returns** – nx2 Matrix of integers. Each row corresponding to a person. The two values of each row indicate the indexes of each person in the people matrix of Population class.

- **removeperson**

```
public void removeperson(int [] identity)
```

- **Description**

Method to remove a person from a place by reducing the People matrix by removing position of the person being added on the population matrix on the Town class.

- **Parameters**

- * **identity** – Couple of integers indicating the position of the removed person in the Population matrix in the Town class.

- **setsize**

```
public void setsize(double size)
```

- **Description**

Method used to set the size of a place.

- **Parameters**

- * **size** – Double indicating the size of the place.

A.6 Class Population

Class representing the population of Oñati. Social groups, family groups, homes, work-places and age will be allocated in this class according to demographic data of Oñati.

A.6.1 Declaration

```
public class Population
    extends java.lang.Object
```

A.6.2 Constructor summary

`Population()`

A.6.3 Method summary

`addtotrack(int[])`
`definetrackingmethod()`
`doDWfamilyactivity()`
`doDWsocialactivity()`
`doindividualleisureactivity()`
`doWEfamilyactivity()`
`doWESocialactivity()`
`getalternativeworkplace(int, int)`
`getcases()`
`getinfected(int[])`
`hometime()`
`infectpeople()`
`isrelevant(int[])`
`leisuretime()`
`passingday()`
`refreshpositions(int[])`
`secondtest(int[])`
`sendhome1()`
`sendhome2()`
`sendtowork1()`
`sendtowork2()`
`setfamilies()`
`sethomes()`
`setmasked(double)`
`setschools()`
`setsocialgroups()`
`setworkplaces()`
`shuffle(int[])`
`simulatestep(int[], int)`
`testcontacts()`
`timeiteration()`
`weekendhometime()`
`weekendleisuretime()`
`worktime()`

A.6.4 Constructors

- Population

```
public Population()
```

– **Description**

Method that constructs the population of Oñati. The population is sorted in a matrix where the column number represents their age. Each person receives social group, a home and a family. For people whose age is between 0 years and 24 years (both inclusive), an education center will be allocated. Each person whose age is between 25 years and 61 years (both inclusive), a workplace will be allocated.

A.6.5 Methods

- **addtotrack**

```
public void addtotrack(int [] identity)
```

– **Description**

Method used to add a person's relatives into tracking lists. Depending on the tracking method, relatives will be added into Confined or Totest peoplelists. People who is already in the "Testing" list won't be added into the Totest list.

– **Parameters**

* `identity` – Id of the person being added to the tracking list

- **definetrackingmethod**

```
public void definetrackingmethod()
```

– **Description**

Method used to define the illness tracking method. A person can be added to a Confined peoplelist or Totest list.

People in the Totest will be confined and tested the next day after introduced in the list, if tested positive, they will remain confined, if tested negative, they will get back to their normal behaviour.

People in the Confined peoplelist will also be confined and tested the next day after they are introduced in the list, however, after the first test they remain confined, until a second test has taken place. To exit the Confined Peoplelist, all members of the same family (shared home) must have tested negative.

- **doDWfamilyactivity**

```
public void doDWfamilyactivity()
```

– **Description**

Method allocate each person a place according to a family activity which is being practiced with people with the same family group.

• **doDWsocialactivity**

```
public void doDWsocialactivity ()
```

– **Description**

Method allocate each person a place according to a group activity which is being practiced with people with the same social group.

• **doindividualeisureactivity**

```
public void doindividualeisureactivity ()
```

– **Description**

Method allocate each person a place to do an individual activity according to the current behaviour profile on the Conduct class.

• **doWEfamilyactivity**

```
public void doWEfamilyactivity ()
```

– **Description**

Method allocate each person a place according to a family activity which is being practiced with people with the same family group.

• **doWEsocialactivity**

```
public void doWEsocialactivity ()
```

– **Description**

Method allocate each person a place according to a group activity which is being practiced with people with the same social group.

• **getalternativeworkplace**

```
public int [] getalternativeworkplace (int i1 ,int i2)
```

– **Description**

Method used to get the common zone of the working place.

- **Parameters**

- * `i1` – Age of the person (position in the population matrix)
- * `i2` – Position of the person in the population matrix.

- **Returns** – Array of 3 elements indicating the common zone of its workplace.

- **getcases**

```
public void getcases ()
```

- **getinfected**

```
public int getinfected (int [] place)
```

- **Description**

Method use to count how many contagious people are in the indicated place.

- **Parameters**

- * `place` – Array indicating the place that is wanted to be evaluated.

- **Returns** – Number of contagious people in the indicated place.

- **hometime**

```
public void hometime ()
```

- **Description**

Method that simulates 8 hours of home time.

- **infectpeople**

```
public void infectpeople ()
```

- **Description**

Method used to infect 5 random people.

- **isrelevant**

```
public boolean isrelevant (int [] place)
```

- **Description**

Method use to determine if a place is relevant to the simulation. A relevant place is defined as a place that has a non zero probability for an infection to happen. For a place to be relevant has to have at least one healthy person and a contagious person.

- **Parameters**

- * `place` – Array indicating the place that is wanted to be evaluated.

- **Returns** – Boolean indicating if the place is relevant or not.

- **leisuretime**

```
public void leisuretime ()
```

- **Description**

- Method that simulates 8 hours of leisure time.

- **passingday**

```
public void passingday ()
```

- **Description**

- Method used to simulate a passing day. Ill people's countdown is reduced by 1. If the countdown drops down to 0, the infected person advances to the next stage.

- **refreshpositions**

```
public void refreshpositions (int [] place)
```

- **Description**

- Method used refresh the position of all the people in the indicated place.

- **Parameters**

- * `place` – Array indicating the place for which the position refreshing is wanted.

- **secondtest**

```
public void secondtest (int [] identity)
```

- **Description**

- Method that simulates a second test. When a person overcomes the illness all of the members of it's family are tested a second time. If all members of the family test negative, they are removed from the Confined list and they return to their normal behaviour.

- **Parameters**

- * `identity` – `Int[]` indicating the person who has overcome the illness.

- **sendhome1**

```
public void sendhome1()
```

- **Description**

Method used to send all the population to their respective homes.

- **sendhome2**

```
public void sendhome2()
```

- **Description**

Method used to send all the population to their respective homes. The difference between this method and sendhome1 is that in this method detected cases are put in quarantine.

- **sendtwork1**

```
public void sendtwork1()
```

- **Description**

Method used to send all the population to their respective workplaces.

- **sendtwork2**

```
public void sendtwork2()
```

- **Description**

Method used to send all the population to their respective workplaces. The difference between this method and sendtwork1 is that in this method people are send to the common zones of their respective workplaces, ej. restrooms, cafeteria...

- **setfamilies**

```
public void setfamilies()
```

- **Description**

Method to give each person a family group. Each family group consists of people from 2-5 different families (people who share a home). Families are assigned a family group randomly.

- **sethomes**

public void sethomes ()

- **Description**

Method to give each person a home. There are 3 types of houses:

- 4 people houses: Homes with 2 adults and 2 children/young people. The age difference between the adults is 2-3 years, same for children. The age difference between adults and children/young people is 30 years.

- 3 people houses: Homes with 2 adults and 1 child or young person. Adults are the same age and the difference between adults and the child / young person is 30 years.

- 2 people houses: Homes with a couple of older people of same age.

People between 0-59 years live in 4 and 3 people houses, whereas people older than 60 live in 2 people houses.

- **setmasked**

public void setmasked(**double** ratio)

- **Description**

Method used to get the common zone of the working place.

- **Parameters**

- * **ratio** – Percentage of people wearing a mask.

- **setschools**

public void setschools ()

- **Description**

Method that gives each person between 0 and 23 years a study center. Each school receives 6 different generations (from 0 to 5, from 6 to 11, from 12 to 17 and from 18 to 23), and each generation is divided into 3 schools, therefore making 12 schools in total. Each school has 2 classrooms per generation, with 20 students in each classroom, making 12 classrooms per school.

- **setsocialgroups**

public void setsocialgroups ()

- **Description**

Method that gives each person a social group. Each generation has 10 social groups.

- **setworkplaces**

```
public void setworkplaces ()
```

- **Description**

Method to give each person between 25 and 61 years a workplace. The method is divided in two parts:

Firstly, 4930 x 3 matrix is built, each row representing a particular job. How many people are assigned each work is obtained from the demographic data of Oñati.

Lastly, each person is given a workplace from the previously constructed matrix. The allocation process is made in a way all works are evenly distributed between people of different age.

People without a job are assigned their home as their "workplace" (place where they stay while worktime).

- **shuffle**

```
public int [] shuffle (int [] array)
```

- **Description**

Method use to shuffle an array randomly.

- **Parameters**

* **array** – Array which is wanted to be shuffled.

- **Returns** – Shuffled array.

- **simulatestep**

```
public void simulatestep (int [] place ,int riskfactor)
```

- **Description**

Method used to simulate a time step in a certain place.

- **Parameters**

* **place** – Array of integers which indicates the simulating place.

* **riskfactor** – Integer indicating the risk factor of the simulating place.

- **testcontacts**

```
public void testcontacts()
```

- **Description**

Method which tests all the people in the Totest list. Positive cases are added to the tracking list and the relatives of the new detected cases are added to the next day's Totest list.

- **timeiteration**

```
public void timeiteration()
```

- **Description**

Method used simulate 1 minute of the simulation. Only active places are simulated. Each person's position is refreshed and for the uninfected people the probability of infection is calculated.

- **weekendhometime**

```
public void weekendhometime()
```

- **Description**

Method that simulates 12 hours of home time.

- **weekendleisuretime**

```
public void weekendleisuretime()
```

- **Description**

Method that simulates 12 hours of leisure time on weekends.

- **worktime**

```
public void worktime()
```

- **Description**

Method that simulates 8 hours of home time.

A.7 Class Town

Method that constructs the town of Oñati. Physical places are divided into 8 groups using several Place class objects. Each place group has an implicit integer associated to it:

- 0: Homes
- 1: Big factories
- 2: Small businesses or administrative offices
- 3: Schools and college
- 4: Bars and restaurants
- 5: Shops
- 6: Health center
- 7: Public places such as parks, sport centers, street...

A.7.1 Declaration

```
public class Town
    extends java.lang.Object
```

A.7.2 Constructor summary

[Town\(\)](#)

A.7.3 Method summary

```
addperson\(int\[\], int\[\]\)
emptyplaces\(\)
getlistofpeople\(int\[\]\)
getsize\(int\[\]\)
removeperson\(int\[\], int\[\]\)
```

A.7.4 Constructors

- **Town**

```
public Town()
```

A.7.5 Methods

- **addperson**

```
public void addperson(int [] place, int [] pointer)
```

- **Description**

Method that adds the indicated person to the indicated place.

- **Parameters**

- * **place** – Array indicating the place where the person is wanted.
- * **pointer** – Pointer indicating the place in the Town class

- **emptyplaces**

```
public void emptyplaces()
```

- **Description**

Method that empties all the places in Town.

- **getlistofpeople**

```
public int [][] getlistofpeople(int [] place)
```

- **Description**

Method which gives a list of people who are in the indicated place.

- **Parameters**

- * **place** – Array indicating the place that is wanted to be evaluated.

- **Returns** – List of people who are in the indicated place.

- **getsize**

```
public double getsize(int [] place)
```

- **Description**

Method which gives the size of the indicated place.

- **Parameters**

- * **place** – Array indicating the place for which the size is asked.

- **Returns** – Size of the indicated place.

- **removeperson**

```
public void removeperson(int [] place, int [] identity)
```

- **Description**

Method that removes the indicated person from the indicated place.

- **Parameters**

- * **place** – Array indicating the place where the person is taken out.
- * **identity** – Couple of integers indicating the position of the removed person in the Population matrix in the Town class.