

TRENTO PARTECIPA - D2

D'Angiò Enea, Mattarolo Alessandro, Nedeljkovic Ivan

January 2026

Indice

1	Introduzione	3
2	Web APIs	4
3	Implementazione	50
3.1	Organizzazione delle repositories	50
3.2	Branching strategy e organizzazione del lavoro	51
3.2.1	Suddivisione Logica del Lavoro	51
3.3	Dipendenze	52
3.3.1	Backend	52
3.3.2	Frontend	53
3.4	Database	53
3.4.1	Categoria	53
3.4.2	Piattaforma	54
3.4.3	Utente	54
3.4.4	Pre_Autorizzato	54
3.4.5	Notifica	55
3.4.6	Bilancio Partecipativo	55
3.4.7	Iniziativa	55
3.4.8	Opzioni Bilancio	56
3.4.9	Voti Bilancio	56
3.4.10	Firma Iniziativa	56
3.4.11	Iniziativa Salvata	57
3.4.12	Risposta	57
3.4.13	Allegato	57
3.5	Testing	58

4	FrontEnd	68
4.1	Dettaglio dell'header	68
4.2	Home Page	69
4.3	Schermata di login	70
4.4	Schermata di creazione di un nuovo profilo	71
4.5	Schermata di creazione di una nuova iniziativa	72
4.6	Schermata di visualizzazione delle iniziative	73
4.7	Schermata di visualizzazione della singola iniziativa	75
4.8	Dashboard personale	76
4.9	Area Admin	77
4.10	Schermata di monitoraggio delle iniziative	78
4.11	Schermata di compilazione di una risposta del Comune	79
4.12	Schermata di creazione del bilancio partecipativo	81
4.13	Schermata di gestione del personale	82
4.14	Archivio dei bilanci partecipativi	83
5	Deployment	84
5.1	Accesso docenti	84

1 Introduzione

Il seguente documento riporta le informazioni relative all'**implementazione** del progetto "TRENTO PARTECIPA".

È importante premettere che alcune funzionalità descritte nel documento D1 non sono state completamente implementate per vari motivi, principalmente legati a fattori di semplicità e tempistiche. In particolare:

- Sebbene i requisiti funzionali prevedano l'utilizzo di SPID o CIE per il login, in questa fase di sviluppo abbiamo optato per sostituirli con l'**accesso tramite Google**. Tale scelta è motivata principalmente da vincoli di natura burocratica: l'integrazione in ambiente di produzione di SPID/CIE richiede infatti che l'ente gestore dell'applicazione ottenga la qualifica formale di Service Provider accreditato presso l'AgID o il Ministero dell'Interno. Questo iter comporta oneri economici, tempistiche burocratiche e requisiti legali non gestibili dal nostro gruppo. Abbiamo quindi deciso di usare l'accesso tramite Google, per simulare comunque l'accesso tramite un identity provider esterno, mentre l'integrazione con SPID/CIE potrà eventualmente essere realizzata in futuro. La conseguenza di questa scelta, in termini di funzionalità, è il fatto che non è più possibile verificare la residenza di un utente durante l'atto di creazione di un profilo. Dunque, a meno di operazioni hardcoded e cambiamenti manuali nel database, qualunque utente non pre-autorizzato che fa il primo accesso all'applicativo ottiene il ruolo di "Cittadino".
- Non è stato implementato l'algoritmo di **controllo duplicati** nella fase di creazione di un'iniziativa.
- L'**importazione delle iniziative** esterne è stata simulata utilizzando delle iniziative demo all'interno di un file *external_source.json*. Nel progetto completo dovrà essere implementato un *algoritmo* capace di raccogliere le iniziative presenti su piattaforme esterne riguardanti il Comune di Trento, formattare e modificare il suddetto file json. L'implementazione attuale utilizza una libreria di *node* per controllare ogni 24 ore il file contenente le iniziative esterne ed importarle nel database. Inoltre è attualmente attiva la funzionalità, utilizzata principalmente per debugging, che permette all'importazione di essere eseguita dopo 10 secondi dall'avvio del server.
- L'**aggiornamento prioritario** è una funzionalità fortemente legata all'*algoritmo per le importazioni* delle iniziative esterne e sarebbe stato efficace esclusivamente se esso fosse stato pienamente implementato. Per questo motivo abbiamo deciso di non procedere con questa piccola funzionalità, indispensabile però nella versione completa del progetto.

2 Web APIs

Le specifiche delle API sono state documentate secondo lo standard OpenAPI 3.0.1 e la documentazione è consultabile pubblicamente al seguente indirizzo: <https://380vqprxjy.apidog.io>

Progettazione e scelte architetturali L'architettura segue i principi RESTful, organizzando le risorse in modo gerarchico. In fase di progettazione è stato deciso di standardizzare le risposte di errore, ottimizzare la gestione delle collezioni (tramite l'utilizzo di metadati per distinguere la pagina ed il numero di elementi necessari) e utilizzare ampiamente schemi ed ereditarietà dei componenti (abbiamo per esempio creato versioni standard e dettagliate per vari schemi in modo da ridurre i dati inutilizzati). Abbiamo inoltre fatto attenzione nell'aggiungere regole di validazione all'interno schemi delle api, come per esempio l'uso di enum per gli stati delle iniziative.

Le specifiche delle APIs sono disponibili al seguente indirizzo: <https://github.com/Neo-org/TrentoPartecipa/blob/main/server/APIsDocumentazione/AllAPIs.yaml> e sono riportate di seguito:

```
1 openapi: 3.0.1
2 info:
3   title: Default module
4   description: ''
5   version: 1.0.0
6 tags:
7   - name: Initiatives
8   - name: ParticipatoryBudgets
9   - name: Users
10  - name: Users/Auth
11  - name: Filters
12 paths:
13   /initiatives:
14     get:
15       summary: initiativesList
16       deprecated: false
17       description: >-
18         Restituisce la lista delle iniziative. Supporta parametri per
19         l'applicazione di filtri alla ricerca, per la ricerca testuale e
20         l'ordinamento.
21       tags:
22         - Initiatives
23       parameters:
24         - name: currentPage
25           in: query
26           description: Numero della pagina per la paginazione.
27           required: false
28           example: 1
29           schema:
30             type: integer
```

```

31 - name: objectsPerPage
32   in: query
33   description: Numero di elementi per pagina.
34   required: false
35   example: 10
36   schema:
37     type: integer
38 - name: status
39   in: query
40   description: Filtro per lo stato dell'iniziativa.
41   required: false
42   example: In corso
43   schema:
44     type: string
45 - name: search
46   in: query
47   description: Ricerca libera nel titolo, descrizione e luogo.
48   required: false
49   example: Parco
50   schema:
51     type: string
52 - name: categoryID
53   in: query
54   description: ID univoco della categoria (filtro per categoria).
55   required: false
56   example: Sport
57   schema:
58     type: integer
59 - name: sortBy
60   in: query
61   description: Codice relativo ad un attributo su cui bisogna ordinare.
62   required: false
63   example: 1
64   schema:
65     type: integer
66 - name: order
67   in: query
68   description: La direzione dell'ordinamento (asc o desc).
69   required: false
70   example: desc
71   schema:
72     type: string
73     enum:
74       - asc
75       - desc
76 - name: minSignatures
77   in: query
78   description: Limite inferiore di firme da mostrare.
79   required: false
80   example: 0
81   schema:
82     type: integer
83 - name: maxSignatures

```

```

84     in: query
85     description: Limite superiore di firme da mostrare.
86     required: false
87     example: 1000
88     schema:
89         type: integer
90 responses:
91     '200':
92         description: ''
93         content:
94             application/json:
95                 schema:
96                     type: object
97                     properties:
98                         data:
99                             type: array
100                             items:
101                                 $ref: '#/components/schemas/Initiative'
102                             minItems: 100
103                             maxItems: 100
104                         meta:
105                             $ref: '#/components/schemas/MetaDataList'
106                     required:
107                         - data
108                         - meta
109         headers: {}
110     '400':
111         description: ''
112         content:
113             application/json:
114                 schema:
115                     $ref: '#/components/schemas/GenericError'
116         headers: {}
117 security: []
118 post:
119     summary: createInitiative
120     deprecated: false
121     description: Permette a un cittadino autenticato di creare una nuova iniziativa.
122     tags:
123         - Initiatives
124     parameters:
125         - name: X-Mock-User-Id
126           in: header
127           description: Simula l'utente loggato
128           required: true
129           example: 1
130           schema:
131               type: integer
132     requestBody:
133         content:
134             multipart/form-data:
135                 schema:
136                     type: object

```

```

137         properties:
138             title:
139                 description: Titolo dell'iniziativa.
140                 type: string
141                 example: ''
142             description:
143                 description: Descrizione dettagliata della proposta.
144                 type: string
145                 example: ''
146             place:
147                 description: Luogo o area di interesse.
148                 type: string
149                 example: ''
150             categoryId:
151                 type: integer
152                 description: ID univoco della categoria.
153                 example: 0
154             attachments:
155                 format: binary
156                 type: string
157                 description: Eventuali allegati (Immagini o PDF).
158                 example: ''
159         required: true
160     responses:
161         '201':
162             description: ''
163             content:
164                 application/json:
165                     schema:
166                         $ref: '#/components/schemas/DetailedInitiative'
167             headers: {}
168         '400':
169             description: ''
170             content:
171                 application/json:
172                     schema:
173                         $ref: '#/components/schemas/GenericError'
174             headers: {}
175         '401':
176             description: ''
177             content:
178                 application/json:
179                     schema:
180                         $ref: '#/components/schemas/GenericError'
181             headers: {}
182         '403':
183             description: ''
184             content:
185                 application/json:
186                     schema:
187                         $ref: '#/components/schemas/GenericError'
188             headers: {}
189         '409':

```

```

190     description: ''
191     content:
192       application/json:
193         schema:
194           type: object
195           properties: {}
196     headers: {}
197   '422':
198     description: ''
199     content:
200       application/json:
201         schema:
202           $ref: '#/components/schemas/GenericError'
203     headers: {}
204   security: []
205 /initiatives/{id}:
206   get:
207     summary: initiativeDetails
208     deprecated: false
209     description: >-
210       Restituisce i dettagli completi di una singola iniziativa selezionata
211       dall'utente.
212     tags:
213       - Initiatives
214     parameters:
215       - name: id
216         in: path
217         description: ID univoco dell'iniziativa.
218         required: true
219         example: 0
220         schema:
221           type: integer
222     responses:
223       '200':
224         description: ''
225         content:
226           application/json:
227             schema:
228               $ref: '#/components/schemas/DetailedInitiative'
229         headers: {}
230       '400':
231         description: ''
232         content:
233           application/json:
234             schema:
235               $ref: '#/components/schemas/GenericError'
236         headers: {}
237       '404':
238         description: ''
239         content:
240           application/json:
241             schema:
242               $ref: '#/components/schemas/GenericError'

```



```

243     headers: {}
244     security: []
245 patch:
246     summary: changeExpirationDate
247     deprecated: false
248     description: >-
249         Permette all'amministrazione di prorogare la data di scadenza di
250         un'iniziativa.
251     tags:
252         - Initiatives
253     parameters:
254         - name: id
255           in: path
256           description: ID univoco dell'iniziativa.
257           required: true
258           example: 0
259           schema:
260             type: integer
261         - name: X-Mock-User-Id
262           in: header
263           description: ''
264           required: true
265           example: 0
266           schema:
267             type: integer
268     requestBody:
269       content:
270         application/json:
271           schema:
272             type: object
273             properties:
274               expirationDate:
275                 type: string
276                 format: date
277                 description: Nuova data di scadenza.
278             required:
279               - expirationDate
280       required: true
281     responses:
282       '200':
283         description: ''
284         content:
285           application/json:
286             schema:
287               $ref: '#/components/schemas/DetailedInitiative'
288             headers: {}
289       '401':
290         description: ''
291         content:
292           application/json:
293             schema:
294               $ref: '#/components/schemas/GenericError'
295             headers: {}

```

```

296     '403':
297         description: ''
298         content:
299             application/json:
300                 schema:
301                     $ref: '#/components/schemas/GenericError'
302         headers: {}
303     '404':
304         description: ''
305         content:
306             application/json:
307                 schema:
308                     $ref: '#/components/schemas/GenericError'
309         headers: {}
310     '422':
311         description: ''
312         content:
313             application/json:
314                 schema:
315                     $ref: '#/components/schemas/GenericError'
316         headers: {}
317     security: []
318 /initiatives/{id}/responses:
319     post:
320         summary: createReply
321         deprecated: false
322         description: >-
323             Permette all'amministrazione di inviare una risposta ufficiale a
324             un'iniziativa, cambiandone lo stato da 'In corso' a 'Respinta' oppure
325             'Approvata'.
326         tags:
327             - Initiatives
328         parameters:
329             - name: id
330               in: path
331               description: ID univoco dell'iniziativa.
332               required: true
333               example: 0
334               schema:
335                 type: integer
336             - name: X-Mock-User-Id
337               in: header
338               description: ''
339               required: true
340               example: 0
341               schema:
342                 type: integer
343         requestBody:
344             content:
345                 multipart/form-data:
346                     schema:
347                         type: object
348                         properties:

```

```

349         status:
350             description: Nuovo stato dell'iniziativa ('Respinta' o 'Approvata').
351             type: string
352             example: ''
353         motivations:
354             description: Testo di esposizione delle motivazioni della risposta.
355             type: string
356             example: ''
357         attachments:
358             format: binary
359             type: string
360             description: Eventuali allegati (immagini o PDF).
361             example: ''
362     required: true
363     responses:
364         '201':
365             description: ''
366             content:
367                 application/json:
368                     schema:
369                         $ref: '#/components/schemas/Reply'
370             headers: {}
371         '400':
372             description: ''
373             content:
374                 application/json:
375                     schema:
376                         $ref: '#/components/schemas/GenericError'
377             headers: {}
378         '401':
379             description: ''
380             content:
381                 application/json:
382                     schema:
383                         $ref: '#/components/schemas/GenericError'
384             headers: {}
385         '403':
386             description: ''
387             content:
388                 application/json:
389                     schema:
390                         $ref: '#/components/schemas/GenericError'
391             headers: {}
392         '409':
393             description: ''
394             content:
395                 application/json:
396                     schema:
397                         $ref: '#/components/schemas/GenericError'
398             headers: {}
399     security: []
400 /initiatives/{id}/signatures:
401     post:

```

```

402     summary: signInitiative
403     deprecated: false
404     description: Permette a un cittadino di firmare un'iniziativa.
405     tags:
406       - Initiatives
407     parameters:
408       - name: id
409         in: path
410         description: ID univoco dell'iniziativa.
411         required: true
412         example: 0
413         schema:
414           type: integer
415       - name: X-Mock-User-Id
416         in: header
417         description: ''
418         required: true
419         example: 0
420         schema:
421           type: integer
422     responses:
423       '201':
424         description: ''
425         content:
426           application/json:
427             schema:
428               $ref: '#/components/schemas/InitiativeSignature'
429         headers: {}
430       '401':
431         description: ''
432         content:
433           application/json:
434             schema:
435               $ref: '#/components/schemas/GenericError'
436         headers: {}
437       '403':
438         description: ''
439         content:
440           application/json:
441             schema:
442               $ref: '#/components/schemas/GenericError'
443         headers: {}
444       '404':
445         description: ''
446         content:
447           application/json:
448             schema:
449               $ref: '#/components/schemas/GenericError'
450         headers: {}
451       '409':
452         description: ''
453         content:
454           application/json:

```

```

455         schema:
456             $ref: '#/components/schemas/GenericError'
457         headers: {}
458         security: []
459 /initiatives/{id}/follows:
460     post:
461         summary: saveIniziativa
462         deprecated: false
463         description: >-
464             Salva un'iniziativa nella lista delle iniziative seguite dall'utente
465             affinché possa monitorarne lo stato.
466         tags:
467             - Initiatives
468         parameters:
469             - name: id
470               in: path
471               description: ID univoco dell'iniziativa.
472               required: true
473               example: 0
474               schema:
475                 type: integer
476             - name: X-Mock-User-Id
477               in: header
478               description: ''
479               required: true
480               example: 0
481               schema:
482                 type: integer
483         responses:
484             '200':
485                 description: ''
486                 content:
487                     application/json:
488                         schema:
489                             $ref: '#/components/schemas/SavedInitiative'
490                 headers: {}
491             '401':
492                 description: ''
493                 content:
494                     application/json:
495                         schema:
496                             $ref: '#/components/schemas/GenericError'
497                 headers: {}
498             '404':
499                 description: ''
500                 content:
501                     application/json:
502                         schema:
503                             $ref: '#/components/schemas/GenericError'
504                 headers: {}
505             '409':
506                 description: ''
507                 content:

```

```

508         application/json:
509             schema:
510                 $ref: '#/components/schemas/GenericError'
511             headers: {}
512         security: []
513 delete:
514     summary: removeFromSavedInitiatives
515     deprecated: false
516     description: Rimuove l'iniziativa dalla lista delle iniziative seguite.
517     tags:
518         - Initiatives
519     parameters:
520         - name: id
521           in: path
522           description: ID univoco dell'iniziativa.
523           required: true
524           example: 0
525           schema:
526               type: integer
527         - name: X-Mock-User-Id
528           in: header
529           description: ''
530           required: true
531           example: 0
532           schema:
533               type: integer
534     responses:
535         '200':
536             description: ''
537             content:
538                 application/json:
539                     schema:
540                         type: object
541                         properties:
542                             message:
543                                 type: string
544                             initiativeId:
545                                 type: integer
546                             required:
547                                 - message
548                                 - initiativeId
549             headers: {}
550         '401':
551             description: ''
552             content:
553                 application/json:
554                     schema:
555                         type: object
556                         properties: {}
557             headers: {}
558         '404':
559             description: ''
560             content:

```

```

561         application/json:
562             schema:
563                 type: object
564                 properties: {}
565             headers: {}
566         security: []
567 /initiatives/admin/expiring:
568 get:
569     summary: getExpiringInitiatives
570     deprecated: false
571     description: >-
572         Restituisce le iniziative in scadenza entro 7 giorni. Solo per
573         amministratori.
574     tags:
575         - Initiatives
576     parameters:
577         - name: currentPage
578           in: query
579           description: Numero della pagina per la paginazione.
580           required: false
581           example: 1
582           schema:
583               type: integer
584         - name: objectsPerPage
585           in: query
586           description: Numero di elementi per pagina.
587           required: false
588           example: 5
589           schema:
590               type: integer
591         - name: X-Mock-User-Id
592           in: header
593           description: Simula l'utente loggato (deve essere admin)
594           required: true
595           example: 1
596           schema:
597               type: integer
598     responses:
599         '200':
600             description: Lista delle iniziative in scadenza.
601             content:
602                 application/json:
603                     schema:
604                         type: object
605                         properties:
606                             data:
607                                 type: array
608                                 items:
609                                     $ref: '#/components/schemas/Initiative'
610                             meta:
611                                 $ref: '#/components/schemas/MetaDataList'
612                         required:
613                             - data

```

```

614         - meta
615         headers: {}
616     '401':
617         description: Utente non autenticato.
618         content:
619             application/json:
620                 schema:
621                     $ref: '#/components/schemas/GenericError'
622         headers: {}
623     '403':
624         description: L'utente non ha privilegi di amministratore.
625         content:
626             application/json:
627                 schema:
628                     $ref: '#/components/schemas/GenericError'
629         headers: {}
630     '500':
631         description: Errore del server.
632         content:
633             application/json:
634                 schema:
635                     $ref: '#/components/schemas/GenericError'
636         headers: {}
637     security: []
638 /initiatives/cooldown:
639     get:
640         summary: checkCooldown
641         deprecated: false
642         description: >-
643             Verifica se l'utente può creare una nuova iniziativa (controlla il
644             periodo di cooldown di 14 giorni dall'ultima iniziativa creata).
645         tags:
646             - Initiatives
647         parameters:
648             - name: X-Mock-User-Id
649               in: header
650               description: Simula l'utente loggato
651               required: true
652               example: 1
653               schema:
654                 type: integer
655         responses:
656             '200':
657                 description: Informazioni sul cooldown.
658                 content:
659                     application/json:
660                         schema:
661                             type: object
662                             properties:
663                                 canCreate:
664                                     type: boolean
665                                     description: Indica se l'utente può creare una nuova iniziativa.
666                                 daysRemaining:

```



```

667         type: integer
668         description: >-
669             Numero di giorni rimanenti prima di poter creare una nuova
670             iniziativa.
671         nullable: true
672         lastInitiativeDate:
673             type: string
674             format: date
675             description: Data dell'ultima iniziativa creata.
676             nullable: true
677         required:
678             - canCreate
679         headers: {}
680     '401':
681         description: Utente non autenticato.
682         content:
683             application/json:
684                 schema:
685                     $ref: '#/components/schemas/GenericError'
686         headers: {}
687     '500':
688         description: Errore del server.
689         content:
690             application/json:
691                 schema:
692                     $ref: '#/components/schemas/GenericError'
693         headers: {}
694     security: []
695 /participatory-budgets:
696     post:
697         summary: createParticipatoryBudget
698         deprecated: false
699         description: Creazione di un nuovo bilancio partecipativo.
700         tags:
701             - ParticipatoryBudgets
702         parameters:
703             - name: X-Mock-User-Id
704               in: header
705               description: ''
706               required: true
707               example: 0
708               schema:
709                 type: integer
710         requestBody:
711             content:
712                 application/json:
713                     schema:
714                         $ref: '#/components/schemas/ParticipatoryBudget'
715             required: true
716         responses:
717             '201':
718                 description: ''
719                 content:

```

```

720     application/json:
721       schema:
722         type: object
723         properties:
724           options:
725             type: array
726             items:
727               $ref: '#/components/schemas/PBOption'
728             minItems: 3
729             maxItems: 3
730             description: Array di opzioni.
731         id:
732           type: integer
733           description: ID univoco del bilancio partecipativo.
734         creatorId:
735           type: integer
736           description: ID univoco del creatore del bilancio partecipativo.
737         title:
738           type: string
739           description: Titolo del bilancio partecipativo.
740         createdAt:
741           type: string
742           format: date-time
743           description: Data di creazione.
744         expirationDate:
745           type: string
746           format: date
747           description: Data di scadenza.
748         votedOptionId:
749           type: integer
750           description: ID dell'opzione votata.
751         required:
752           - votedOptionId
753           - options
754           - id
755           - creatorId
756           - title
757           - createdAt
758           - expirationDate
759     headers: {}
760 '400':
761   description: ''
762   content:
763     application/json:
764       schema:
765         type: object
766         properties: {}
767     headers: {}
768 '401':
769   description: ''
770   content:
771     application/json:
772       schema:

```

```

773         $ref: '#/components/schemas/GenericError'
774     headers: {}
775     '403':
776         description: ''
777         content:
778             application/json:
779                 schema:
780                     $ref: '#/components/schemas/GenericError'
781         headers: {}
782     '409':
783         description: ''
784         content:
785             application/json:
786                 schema:
787                     $ref: '#/components/schemas/GenericError'
788         headers: {}
789     security: []
790 get:
791     summary: participatoryBudgetsArchive
792     deprecated: false
793     description: Restituisce l'archivio storico dei bilanci partecipativi conclusi.
794     tags:
795         - ParticipatoryBudgets
796     parameters:
797         - name: currentPage
798           in: query
799           description: Numero della pagina per la paginazione.
800           required: false
801           example: 1
802           schema:
803               type: integer
804         - name: objectsPerPage
805           in: query
806           description: Numero di elementi per pagina.
807           required: false
808           example: 10
809           schema:
810               type: integer
811         - name: X-Mock-User-id
812           in: header
813           description: ''
814           required: true
815           example: 0
816           schema:
817               type: integer
818     responses:
819         '200':
820             description: ''
821             content:
822                 application/json:
823                     schema:
824                         type: object
825                         properties:

```

```

826         data:
827             type: array
828             items:
829                 $ref: '#/components/schemas/ParticipatoryBudget'
830             minItems: 1
831             maxItems: 5
832         meta:
833             $ref: '#/components/schemas/MetaDataList'
834         required:
835             - data
836             - meta
837         headers: {}
838     '401':
839         description: ''
840         content:
841             application/json:
842                 schema:
843                     $ref: '#/components/schemas/GenericError'
844         headers: {}
845     '403':
846         description: ''
847         content:
848             application/json:
849                 schema:
850                     $ref: '#/components/schemas/GenericError'
851         headers: {}
852     security: []
853 /participatory-budgets/active:
854     get:
855         summary: participatoryBudgetDetails
856         deprecated: false
857         description: Restituisce i dettagli di un bilancio partecipativo.
858         tags:
859             - ParticipatoryBudgets
860         parameters:
861             - name: status
862               in: query
863               description: Filtro per stato (attivo o non).
864               required: false
865               schema:
866                 type: string
867                 enum:
868                     - active
869                     - notActive
870             - name: X-Mock-User-Id
871               in: header
872               description: ''
873               required: true
874               example: 0
875               schema:
876                 type: integer
877         responses:
878             '200':

```

```

879     description: ''
880     content:
881       application/json:
882         schema:
883           type: object
884           properties:
885             votedOptionId:
886               type: integer
887               description: >-
888                 Identifica l'opzione che l'utente ha votato se ha già
889                 votato.
890               nullable: true
891             options:
892               type: array
893               items:
894                 $ref: '#/components/schemas/PBOption'
895               minItems: 3
896               maxItems: 3
897               description: Array di opzioni.
898             id:
899               type: integer
900               description: ID univoco del bilancio partecipativo.
901             creatorId:
902               type: integer
903               description: ID univoco del creatore del bilancio partecipativo.
904             title:
905               type: string
906               description: Titolo del bilancio partecipativo.
907             createdAt:
908               type: string
909               format: date-time
910               description: Data di creazione.
911             expirationDate:
912               type: string
913               format: date
914               description: Data di scadenza.
915             required:
916               - options
917               - id
918               - creatorId
919               - title
920               - createdAt
921               - expirationDate
922           headers: {}
923           security: []
924 /participatory-budgets/{id}/votes:
925   post:
926     summary: voteParticipatoryBudget
927     deprecated: false
928     description: >-
929       Permette a un cittadino di esprimere il proprio voto per una delle
930       opzioni del bilancio partecipativo.
931     tags:

```

```

932     - ParticipatoryBudgets
933 parameters:
934   - name: id
935     in: path
936     description: ID univoco del bilancio partecipativo.
937     required: true
938     example: 0
939     schema:
940       type: integer
941   - name: X-Mock-User-Id
942     in: header
943     description: ''
944     required: true
945     example: 0
946     schema:
947       type: integer
948 requestBody:
949   content:
950     application/json:
951       schema:
952         type: object
953         properties:
954           position:
955             type: integer
956             minimum: 2
957             maximum: 5
958             description: Posizione dell'opzione scelta.
959             required:
960               - position
961         required: true
962 responses:
963   '200':
964     description: ''
965     content:
966       application/json:
967         schema:
968           $ref: '#/components/schemas/ParticipatoryBudget'
969     headers: {}
970   '400':
971     description: ''
972     content:
973       application/json:
974         schema:
975           $ref: '#/components/schemas/GenericError'
976     headers: {}
977   '401':
978     description: ''
979     content:
980       application/json:
981         schema:
982           $ref: '#/components/schemas/GenericError'
983     headers: {}
984   '403':

```

```

985         description: ''
986         content:
987             application/json:
988                 schema:
989                     $ref: '#/components/schemas/GenericError'
990         headers: {}
991     '404':
992         description: ''
993         content:
994             application/json:
995                 schema:
996                     $ref: '#/components/schemas/GenericError'
997         headers: {}
998     '409':
999         description: ''
1000        content:
1001            application/json:
1002                schema:
1003                    $ref: '#/components/schemas/GenericError'
1004        headers: {}
1005        security: []
1006    /auth/google:
1007        post:
1008            summary: loginAndAuth
1009            deprecated: false
1010            description: >-
1011                Autenticazione utente tramite provider esterno. Verifica l'identità e
1012                restituisce il token di sessione.
1013            tags:
1014                - Users/Auth
1015            parameters: []
1016            requestBody:
1017                content:
1018                    application/json:
1019                        schema:
1020                            type: object
1021                            properties:
1022                                tokenId:
1023                                    type: string
1024                                    description: Token fornito dal provider di identità.
1025                                required:
1026                                    - tokenId
1027            required: true
1028        responses:
1029            '200':
1030                description: ''
1031                content:
1032                    application/json:
1033                        schema:
1034                            $ref: '#/components/schemas/AuthResponse'
1035                headers: {}
1036            '400':
1037                description: ''

```

```

1038     content:
1039       application/json:
1040         schema:
1041           $ref: '#/components/schemas/GenericError'
1042     headers: {}
1043   '401':
1044     description: ''
1045     content:
1046       application/json:
1047         schema:
1048           $ref: '#/components/schemas/GenericError'
1049     headers: {}
1050   '404':
1051     description: ''
1052     content:
1053       application/json:
1054         schema:
1055           title: ''
1056           type: object
1057           properties:
1058             status:
1059               type: string
1060             googleData:
1061               type: object
1062               properties:
1063                 firstName:
1064                   type: string
1065                 lastName:
1066                   type: string
1067                 email:
1068                   type: string
1069                 fiscalCode:
1070                   type: string
1071             required:
1072               - firstName
1073               - lastName
1074               - email
1075               - fiscalCode
1076             required:
1077               - status
1078               - googleData
1079     headers: {}
1080   '500':
1081     description: ''
1082     content:
1083       application/json:
1084         schema:
1085           $ref: '#/components/schemas/GenericError'
1086     headers: {}
1087   security: []
1088 /auth/logout:
1089   post:
1090     summary: logout

```



```

1091 deprecated: false
1092 description: Termina la sessione dell'utente corrente.
1093 tags:
1094   - Users/Auth
1095 parameters: []
1096 responses:
1097   '200':
1098     description: ''
1099     content:
1100       application/json:
1101         schema:
1102           type: object
1103           properties:
1104             message:
1105               type: string
1106           required:
1107             - message
1108         headers: {}
1109     security: []
1110 /auth/otp:
1111   post:
1112     summary: richiestaOTP
1113     deprecated: false
1114     description: >-
1115       Invia un codice OTP (One-Time Password) via email per la verifica
1116       dell'identità durante la registrazione di nuovi utenti.
1117     tags:
1118       - Users/Auth
1119     parameters: []
1120     requestBody:
1121       content:
1122         application/json:
1123           schema:
1124             type: object
1125             properties:
1126               email:
1127                 type: string
1128             required:
1129               - email
1130       required: true
1131     responses:
1132       '201':
1133         description: ''
1134         content:
1135           application/json:
1136             schema:
1137               type: object
1138               properties:
1139                 message:
1140                   type: string
1141             required:
1142               - message
1143         headers: {}

```

```

1144     security: []
1145 /auth/register:
1146   post:
1147     summary: userRegistration
1148     deprecated: false
1149     description: >-
1150       Registrazione di un nuovo utente (Cittadino o Amministratore
1151       pre-autorizzato) dopo la verifica OTP.
1152     tags:
1153       - Users/Auth
1154     parameters: []
1155     requestBody:
1156       content:
1157         application/json:
1158           schema:
1159             type: object
1160             properties:
1161               firstName:
1162                 type: string
1163                 description: Nome dell'utente.
1164               lastName:
1165                 type: string
1166                 description: Cognome dell'utente.
1167               fiscalCode:
1168                 type: string
1169                 description: Codice fiscale dell'utente.
1170               email:
1171                 type: string
1172                 description: Email dell'utente.
1173               otp:
1174                 type: string
1175                 description: Codice OTP ricevuto via email.
1176             required:
1177               - firstName
1178               - lastName
1179               - fiscalCode
1180               - email
1181               - otp
1182             required: true
1183     responses:
1184       '201':
1185         description: Utente registrato con successo.
1186         content:
1187           application/json:
1188             schema:
1189               $ref: '#/components/schemas/AuthResponse'
1190             headers: {}
1191       '400':
1192         description: Richiesta non valida o OTP errato.
1193         content:
1194           application/json:
1195             schema:
1196               $ref: '#/components/schemas/GenericError'

```

```

1197     headers: {}
1198   '409':
1199     description: Utente già esistente.
1200     content:
1201       application/json:
1202         schema:
1203           $ref: '#/components/schemas/GenericError'
1204     headers: {}
1205   '500':
1206     description: Errore del server.
1207     content:
1208       application/json:
1209         schema:
1210           $ref: '#/components/schemas/GenericError'
1211     headers: {}
1212   security: []
1213 /users/me/initiatives:
1214   get:
1215     summary: initiativesDashboard
1216     deprecated: false
1217     description: >-
1218       Restituisce i dati per la dashboard personale dell'utente (iniziative
1219       firmate, create o seguite).
1220     tags:
1221       - Users
1222     parameters:
1223       - name: relation
1224         in: query
1225         description: Tipo di relazione (signed, created, followed).
1226         required: false
1227         example: created
1228         schema:
1229           type: string
1230           enum:
1231             - created
1232             - signed
1233             - followed
1234       - name: currentPage
1235         in: query
1236         description: Numero della pagina per la paginazione.
1237         required: false
1238         schema:
1239           type: integer
1240       - name: objectsPerPage
1241         in: query
1242         description: Numero di elementi per pagina.
1243         required: false
1244         schema:
1245           type: integer
1246       - name: X-Mock-User-Id
1247         in: header
1248         description: ''
1249         required: true

```

```

1250     example: 0
1251     schema:
1252       type: integer
1253   responses:
1254     '200':
1255       description: ''
1256       content:
1257         application/json:
1258           schema:
1259             type: object
1260             properties:
1261               data:
1262                 type: array
1263                 items:
1264                   $ref: '#/components/schemas/DetailedInitiative'
1265               meta:
1266                 $ref: '#/components/schemas/MetaDataList'
1267             required:
1268               - data
1269               - meta
1270           headers: {}
1271     '400':
1272       description: ''
1273       content:
1274         application/json:
1275           schema:
1276             $ref: '#/components/schemas/GenericError'
1277       headers: {}
1278     '401':
1279       description: ''
1280       content:
1281         application/json:
1282           schema:
1283             $ref: '#/components/schemas/GenericError'
1284       headers: {}
1285   security: []
1286 /users/me/notifications:
1287   get:
1288     summary: notificationsList
1289     deprecated: false
1290     description: >-
1291       Restituisce la lista delle notifiche dell'utente loggato, con
1292       possibilità di filtrare per lette/non lette.
1293     tags:
1294       - Users
1295     parameters:
1296       - name: read
1297         in: query
1298         description: Se false, restituisce solo le notifiche non lette.
1299         required: false
1300         example: 'false'
1301         schema:
1302           type: boolean

```

```

1303     - name: currentPage
1304       in: query
1305       description: Numero della pagina per la paginazione.
1306       required: false
1307       schema:
1308         type: integer
1309     - name: objectsPerPage
1310       in: query
1311       description: Numero di elementi per pagina.
1312       required: false
1313       schema:
1314         type: integer
1315     - name: X-Mock-User-Id
1316       in: header
1317       description: ''
1318       required: true
1319       example: 0
1320       schema:
1321         type: integer
1322     - name: ETag
1323       in: header
1324       description: >-
1325         Specifica la versione corrispondente della modifica, serve per
1326         controllare la concorrenza delle risorse alla modifica dello stato.
1327       required: false
1328       example: '"v1"'
1329       schema:
1330         type: string
1331 responses:
1332   '200':
1333     description: ''
1334     content:
1335       application/json:
1336         schema:
1337           type: object
1338           properties:
1339             data:
1340               type: array
1341               items:
1342                 $ref: '#/components/schemas/Notification'
1343             meta:
1344                 $ref: '#/components/schemas/MetaDataList'
1345           required:
1346             - data
1347             - meta
1348         headers: {}
1349   '400':
1350     description: ''
1351     content:
1352       application/json:
1353         schema:
1354           $ref: '#/components/schemas/GenericError'
1355         headers: {}

```

```

1356     '401':
1357         description: ''
1358         content:
1359             application/json:
1360                 schema:
1361                     $ref: '#/components/schemas/GenericError'
1362         headers: {}
1363     security: []
1364 /users/me/notifications/{id}:
1365     patch:
1366         summary: readNotification
1367         deprecated: false
1368         description: Segna una specifica notifica come letta.
1369         tags:
1370             - Users
1371         parameters:
1372             - name: id
1373               in: path
1374               description: ID univoco della notifica.
1375               required: true
1376               example: 0
1377               schema:
1378                 type: integer
1379             - name: X-Mock-User-Id
1380               in: header
1381               description: ''
1382               required: true
1383               example: 0
1384               schema:
1385                 type: integer
1386         requestBody:
1387             content:
1388                 application/json:
1389                     schema:
1390                         type: object
1391                         properties:
1392                             isRead:
1393                                 type: boolean
1394                                 default: true
1395                                 description: Impostare a true per segnare come letta.
1396                             required:
1397                                 - isRead
1398             required: true
1399         responses:
1400             '200':
1401                 description: ''
1402                 content:
1403                     application/json:
1404                         schema:
1405                             $ref: '#/components/schemas/Notification'
1406                 headers: {}
1407             '400':
1408                 description: ''

```

```

1409         content:
1410             application/json:
1411                 schema:
1412                     $ref: '#/components/schemas/GenericError'
1413             headers: {}
1414         '401':
1415             description: ''
1416             content:
1417                 application/json:
1418                     schema:
1419                         $ref: '#/components/schemas/GenericError'
1420             headers: {}
1421         '403':
1422             description: ''
1423             content:
1424                 application/json:
1425                     schema:
1426                         $ref: '#/components/schemas/GenericError'
1427             headers: {}
1428         '404':
1429             description: ''
1430             content:
1431                 application/json:
1432                     schema:
1433                         $ref: '#/components/schemas/GenericError'
1434             headers: {}
1435     security: []
1436 /users/me:
1437     get:
1438         summary: getUser
1439         deprecated: false
1440         description: Restituisce i dettagli del profilo dell'utente loggato.
1441         tags:
1442             - Users
1443         parameters:
1444             - name: X-Mock-User-Id
1445               in: header
1446               description: ''
1447               required: false
1448               example: ''
1449               schema:
1450                 type: string
1451         responses:
1452             '200':
1453                 description: ''
1454                 content:
1455                     application/json:
1456                         schema:
1457                             $ref: '#/components/schemas/User'
1458                 headers: {}
1459             '400':
1460                 description: ''
1461                 content:

```

```

1462         application/json:
1463             schema:
1464                 $ref: '#/components/schemas/GenericError'
1465             headers: {}
1466         '401':
1467             description: ''
1468             content:
1469                 application/json:
1470                     schema:
1471                         $ref: '#/components/schemas/GenericError'
1472             headers: {}
1473         '403':
1474             description: ''
1475             content:
1476                 application/json:
1477                     schema:
1478                         $ref: '#/components/schemas/GenericError'
1479             headers: {}
1480         '404':
1481             description: ''
1482             content:
1483                 application/json:
1484                     schema:
1485                         $ref: '#/components/schemas/GenericError'
1486             headers: {}
1487         security: []
1488 /users:
1489     post:
1490         summary: userRegistration
1491         deprecated: false
1492         description: >-
1493             Registrazione di un nuovo utente (Cittadino o Amministratore) al primo
1494             accesso.
1495         tags:
1496             - Users
1497         parameters: []
1498         requestBody:
1499             content:
1500                 application/json:
1501                     schema:
1502                         type: object
1503                         properties:
1504                             email:
1505                                 type: string
1506                                 description: Email inserita dall'utente.
1507                             googleToken:
1508                                 type: string
1509                             otp:
1510                                 type: string
1511                         required:
1512                             - email
1513                             - googleToken
1514                             - otp

```



```

1515     required: true
1516 responses:
1517   '201':
1518     description: ''
1519     content:
1520       application/json:
1521         schema:
1522           $ref: '#/components/schemas/User'
1523     headers: {}
1524   '400':
1525     description: ''
1526     content:
1527       application/json:
1528         schema:
1529           $ref: '#/components/schemas/GenericError'
1530     headers: {}
1531   '401':
1532     description: ''
1533     content:
1534       application/json:
1535         schema:
1536           $ref: '#/components/schemas/GenericError'
1537     headers: {}
1538   '403':
1539     description: ''
1540     content:
1541       application/json:
1542         schema:
1543           $ref: '#/components/schemas/GenericError'
1544     headers: {}
1545   '409':
1546     description: ''
1547     content:
1548       application/json:
1549         schema:
1550           $ref: '#/components/schemas/GenericError'
1551     headers: {}
1552 security: []
1553 get:
1554   summary: showAdminUsers
1555   deprecated: false
1556   description: >-
1557     Permette agli amministratori di visualizzare e cercare nella lista degli
1558     utenti registrati.
1559   tags:
1560     - Users
1561   parameters:
1562     - name: isAdmin
1563       in: query
1564       description: Filtra per mostrare solo gli admin.
1565       required: false
1566       schema:
1567         type: boolean

```

```

1568     - name: X-Mock-User-Id
1569       in: header
1570       description: ''
1571       required: true
1572       example: 0
1573       schema:
1574         type: integer
1575   responses:
1576     '200':
1577       description: ''
1578       content:
1579         application/json:
1580           schema:
1581             type: object
1582             properties:
1583               data:
1584                 type: array
1585                 items:
1586                   type: object
1587                   properties:
1588                     firstName:
1589                       type: string
1590                     lastName:
1591                       type: string
1592                     fiscalCode:
1593                       type: string
1594                   required:
1595                     - firstName
1596                     - lastName
1597                     - fiscalCode
1598             required:
1599               - data
1600       headers: {}
1601     '401':
1602       description: ''
1603       content:
1604         application/json:
1605           schema:
1606             $ref: '#/components/schemas/GenericError'
1607       headers: {}
1608     '403':
1609       description: ''
1610       content:
1611         application/json:
1612           schema:
1613             $ref: '#/components/schemas/GenericError'
1614       headers: {}
1615   security: []
1616 /users/{id}:
1617   patch:
1618     summary: changePrivileges
1619     deprecated: false
1620     description: Permette di modificare i privilegi di un utente.

```

```

1621     tags:
1622     - Users
1623   parameters:
1624     - name: id
1625       in: path
1626       description: ID univoco dell'utente.
1627       required: true
1628       example: 0
1629       schema:
1630         type: integer
1631     - name: X-Mock-User-Id
1632       in: header
1633       description: ''
1634       required: true
1635       example: ''
1636       schema:
1637         type: string
1638   requestBody:
1639     content:
1640       application/json:
1641         schema:
1642           type: object
1643           properties:
1644             isAdmin:
1645               type: boolean
1646               description: Di default false.
1647             required:
1648               - isAdmin
1649           required: true
1650   responses:
1651     '200':
1652       description: ''
1653       content:
1654         application/json:
1655           schema:
1656             type: object
1657             properties: {}
1658       headers: {}
1659   security: []
1660 /users/me/notifications/mark-all-as-read:
1661   patch:
1662     summary: markAllNotificationsAsRead
1663     deprecated: false
1664     description: Segna tutte le notifiche dell'utente corrente come lette.
1665     tags:
1666     - Users
1667   parameters:
1668     - name: X-Mock-User-Id
1669       in: header
1670       description: Simula l'utente loggato
1671       required: true
1672       example: 1
1673       schema:

```

```

1674         type: integer
1675 requestBody:
1676   content:
1677     application/json:
1678       schema:
1679         type: object
1680         properties: {}
1681 responses:
1682   '200':
1683     description: Tutte le notifiche sono state segnate come lette.
1684     content:
1685       application/json:
1686         schema:
1687           type: object
1688           properties:
1689             message:
1690               type: string
1691             updatedCount:
1692               type: integer
1693             description: Numero di notifiche aggiornate.
1694             required:
1695               - message
1696               - updatedCount
1697             headers: {}
1698   '401':
1699     description: Utente non autenticato.
1700     content:
1701       application/json:
1702         schema:
1703           $ref: '#/components/schemas/GenericError'
1704       headers: {}
1705   '500':
1706     description: Errore del server.
1707     content:
1708       application/json:
1709         schema:
1710           $ref: '#/components/schemas/GenericError'
1711       headers: {}
1712 security: []
1713 /users/admin/pre-authorize:
1714   post:
1715     summary: preAuthorizeAdmin
1716     deprecated: false
1717     description: >-
1718       Pre-autorizza un codice fiscale per la registrazione come
1719       amministratore. Solo gli amministratori possono utilizzare questa API.
1720     tags:
1721       - Users
1722     parameters:
1723       - name: X-Mock-User-Id
1724         in: header
1725         description: Simula l'utente loggato (deve essere admin)
1726         required: true

```

```

1727     example: 1
1728     schema:
1729       type: integer
1730 requestBody:
1731   content:
1732     application/json:
1733       schema:
1734         type: object
1735         properties:
1736           fiscalCode:
1737             type: string
1738             description: Codice fiscale dell'amministratore da pre-autorizzare.
1739             required:
1740               - fiscalCode
1741   required: true
1742 responses:
1743   '201':
1744     description: Amministratore pre-autorizzato con successo.
1745     content:
1746       application/json:
1747         schema:
1748           type: object
1749           properties:
1750             message:
1751               type: string
1752             fiscalCode:
1753               type: string
1754             required:
1755               - message
1756               - fiscalCode
1757         headers: {}
1758   '400':
1759     description: Richiesta non valida.
1760     content:
1761       application/json:
1762         schema:
1763           $ref: '#/components/schemas/GenericError'
1764         headers: {}
1765   '401':
1766     description: Utente non autenticato.
1767     content:
1768       application/json:
1769         schema:
1770           $ref: '#/components/schemas/GenericError'
1771         headers: {}
1772   '403':
1773     description: L'utente non ha privilegi di amministratore.
1774     content:
1775       application/json:
1776         schema:
1777           $ref: '#/components/schemas/GenericError'
1778         headers: {}
1779   '409':

```

```

1780     description: Codice fiscale già pre-autorizzato.
1781     content:
1782       application/json:
1783         schema:
1784           $ref: '#/components/schemas/GenericError'
1785     headers: {}
1786   '500':
1787     description: Errore del server.
1788     content:
1789       application/json:
1790         schema:
1791           $ref: '#/components/schemas/GenericError'
1792     headers: {}
1793   security: []
1794 /categories:
1795   get:
1796     summary: categoriesList
1797     deprecated: false
1798     description: Restituisce la lista delle categorie.
1799     tags:
1800       - Filters
1801     parameters: []
1802     responses:
1803       '200':
1804         description: ''
1805         content:
1806           application/json:
1807             schema:
1808               type: object
1809               properties:
1810                 data:
1811                   type: array
1812                   items:
1813                     $ref: '#/components/schemas/Category'
1814                 required:
1815                   - data
1816             headers: {}
1817       '500':
1818         description: ''
1819         content:
1820           application/json:
1821             schema:
1822               $ref: '#/components/schemas/GenericError'
1823         headers: {}
1824     security: []
1825 /platforms:
1826   get:
1827     summary: platformsList
1828     deprecated: false
1829     description: Restituisce la lista delle piattaforme.
1830     tags:
1831       - Filters
1832     parameters: []

```

```

1833     responses:
1834       '200':
1835         description: ''
1836         content:
1837           application/json:
1838             schema:
1839               type: object
1840               properties:
1841                 data:
1842                   type: array
1843                   items:
1844                     $ref: '#/components/schemas/Platform'
1845                 required:
1846                   - data
1847             headers: {}
1848       '500':
1849         description: ''
1850         content:
1851           application/json:
1852             schema:
1853               $ref: '#/components/schemas/GenericError'
1854         headers: {}
1855     security: []
1856 components:
1857   schemas:
1858     DetailedInitiative:
1859       properties:
1860         id:
1861           description: ID univoco dell'iniziativa.
1862           type: integer
1863           minimum: 1
1864           maximum: 2147483647
1865         title:
1866           description: Titolo dell'iniziativa.
1867           type: string
1868           maxLength: 255
1869         description:
1870           description: Descrizione dettagliata della proposta.
1871           type: string
1872           maxLength: 65535
1873         place:
1874           description: Luogo o area di interesse.
1875           default: 'NULL'
1876           type: string
1877           maxLength: 64
1878           nullable: true
1879         status:
1880           type: string
1881           description: Stato attuale dell'iniziativa.
1882           default: ''In corso''
1883         enum:
1884           - In corso
1885           - Approvata

```

```

1886         - Respinta
1887         - Scaduta
1888         nullable: true
1889     signatures:
1890         description: Numero di firme raccolte.
1891         default: '0'
1892         type: integer
1893         minimum: -2147483648
1894         maximum: 2147483647
1895         nullable: true
1896     creationDate:
1897         type: string
1898         description: Data di creazione.
1899         format: date
1900     expirationDate:
1901         description: Data di scadenza.
1902         default: 'NULL'
1903         type: string
1904         format: date
1905         nullable: true
1906     authorId:
1907         description: 'ID univoco dell''autore. '
1908         default: 'NULL'
1909         type: integer
1910         minimum: -2147483648
1911         maximum: 2147483647
1912         nullable: true
1913     categoryId:
1914         description: ID univoco della categoria.
1915         type: integer
1916         minimum: -2147483648
1917         maximum: 2147483647
1918     platformId:
1919         description: ID univoco della piattaforma di provenienza.
1920         default: 'NULL'
1921         type: integer
1922         minimum: -2147483648
1923         maximum: 2147483647
1924         nullable: true
1925     externalURL:
1926         description: URL di reindirizzamento alla piattaforma esterna.
1927         default: 'NULL'
1928         type: string
1929         maxLength: 500
1930         nullable: true
1931     attachments:
1932         type: array
1933         items:
1934             $ref: '#/components/schemas/Attachment'
1935         description: Array contenente gli eventuali allegati all'iniziativa.
1936         nullable: true
1937     reply:
1938         $ref: '#/components/schemas/Reply'

```



```

1939         description: Eventuale risposta del Comune.
1940         nullable: true
1941     type: object
1942     required:
1943     - id
1944     - title
1945     - description
1946     - creationDate
1947     - categoryId
1948 ParticipatoryBudget:
1949     type: object
1950     properties:
1951     options:
1952     type: array
1953     items:
1954     $ref: '#/components/schemas/PBOption'
1955     minItems: 3
1956     maxItems: 3
1957     description: Array di opzioni.
1958     id:
1959     type: integer
1960     description: ID univoco del bilancio partecipativo.
1961     creatorId:
1962     type: integer
1963     description: ID univoco del creatore del bilancio partecipativo.
1964     title:
1965     type: string
1966     description: Titolo del bilancio partecipativo.
1967     createdAt:
1968     type: string
1969     format: date-time
1970     description: Data di creazione.
1971     expirationDate:
1972     type: string
1973     format: date
1974     description: Data di scadenza.
1975     required:
1976     - options
1977     - id
1978     - creatorId
1979     - title
1980     - createdAt
1981     - expirationDate
1982 AuthResponse:
1983     type: object
1984     properties:
1985     accessToken:
1986     type: string
1987     user:
1988     $ref: '#/components/schemas/User'
1989     newUserId:
1990     type: boolean
1991     required:

```

```

1992     - accessToken
1993     - user
1994     - newUserId
1995 GenericError:
1996   type: object
1997   properties:
1998     timeStamp:
1999       type: string
2000       format: date-time
2001       description: Data e ora dell'errore.
2002     message:
2003       type: string
2004       description: Messaggio dell'errore.
2005     details:
2006       type: array
2007       items:
2008         type: object
2009         properties:
2010           field:
2011             type: string
2012           issue:
2013             type: string
2014           required:
2015             - field
2016             - issue
2017         nullable: true
2018   required:
2019     - timeStamp
2020     - message
2021 MetaDataList:
2022   type: object
2023   properties:
2024     currentPage:
2025       type: integer
2026       description: Numero della pagina per la paginazione.
2027     objectsPerPage:
2028       type: integer
2029       description: Numero di elementi per pagina.
2030     totalObjects:
2031       type: integer
2032       description: Numero totale di oggetti.
2033     totalPages:
2034       type: integer
2035       description: Numero totale di pagine.
2036   required:
2037     - currentPage
2038     - objectsPerPage
2039     - totalObjects
2040     - totalPages
2041 AttachmentNoID:
2042   properties:
2043     fileName:
2044       description: ''

```

```

2045         type: string
2046         maxLength: 255
2047     filePath:
2048         description: ''
2049         type: string
2050         maxLength: 500
2051     fileType:
2052         description: ''
2053         default: 'NULL'
2054         type: string
2055         maxLength: 50
2056         nullable: true
2057     uploadedAt:
2058         description: ''
2059         default: current_timestamp()
2060         type: string
2061 type: object
2062 required:
2063     - fileName
2064     - filePath
2065     - uploadedAt
2066 Attachment:
2067     properties:
2068         id:
2069             description: ID univoco dell'allegato.
2070             type: integer
2071             minimum: 1
2072             maximum: 2147483647
2073         fileName:
2074             description: Nome del file.
2075             type: string
2076             maxLength: 255
2077         filePath:
2078             description: Path del file.
2079             type: string
2080             maxLength: 500
2081         fileType:
2082             description: Tipo di file.
2083             default: 'NULL'
2084             type: string
2085             maxLength: 50
2086             nullable: true
2087         uploadedAt:
2088             description: Data di caricamento.
2089             default: current_timestamp()
2090             type: string
2091 type: object
2092 required:
2093     - id
2094     - fileName
2095     - filePath
2096     - uploadedAt
2097 Category:

```

```

2098     properties:
2099         id:
2100             description: ID univoco della categoria.
2101             type: integer
2102             minimum: 1
2103             maximum: 2147483647
2104         name:
2105             description: Nome della categoria.
2106             type: string
2107             maxLength: 100
2108     type: object
2109     required:
2110         - id
2111         - name
2112 InitiativeSignature:
2113     properties:
2114         userId:
2115             description: ID dell'utente firmatario.
2116             type: integer
2117             minimum: -2147483648
2118             maximum: 2147483647
2119         initiativeId:
2120             description: ID dell'iniziativa firmata.
2121             type: integer
2122             minimum: -2147483648
2123             maximum: 2147483647
2124         signatureDate:
2125             description: Data della firma.
2126             default: current_timestamp()
2127             type: string
2128     type: object
2129     required:
2130         - userId
2131         - initiativeId
2132         - signatureDate
2133 Initiative:
2134     properties:
2135         id:
2136             description: ID univoco dell'iniziativa.
2137             type: integer
2138             minimum: 1
2139             maximum: 2147483647
2140         title:
2141             description: Titolo dell'iniziativa.
2142             type: string
2143             maxLength: 255
2144         place:
2145             description: Luogo o area di interesse.
2146             default: 'NULL'
2147             type: string
2148             maxLength: 64
2149             nullable: true
2150     status:

```

```

2151     type: string
2152     description: Stato attuale dell'iniziativa.
2153     default: "'In corso'"
2154     enum:
2155         - In corso
2156         - Approvata
2157         - Respinta
2158         - Scaduta
2159     nullable: true
2160 signatures:
2161     description: Numero di firme raccolte.
2162     default: '0'
2163     type: integer
2164     minimum: -2147483648
2165     maximum: 2147483647
2166     nullable: true
2167 creationDate:
2168     type: string
2169     description: Data di creazione.
2170     format: date
2171 expirationDate:
2172     description: Data di scadenza.
2173     default: 'NULL'
2174     type: string
2175     format: date
2176     nullable: true
2177 authorId:
2178     description: 'ID univoco dell''autore. '
2179     default: 'NULL'
2180     type: integer
2181     minimum: -2147483648
2182     maximum: 2147483647
2183     nullable: true
2184 categoryId:
2185     description: ID univoco della categoria.
2186     type: integer
2187     minimum: -2147483648
2188     maximum: 2147483647
2189 platformId:
2190     description: ID univoco della piattaforma.
2191     default: 'NULL'
2192     type: integer
2193     minimum: -2147483648
2194     maximum: 2147483647
2195     nullable: true
2196 externalURL:
2197     description: URL di reindirizzamento alla piattaforma esterna.
2198     default: 'NULL'
2199     type: string
2200     maxLength: 500
2201     nullable: true
2202 attachment:
2203     $ref: '#/components/schemas/Attachment'

```

```

2204         description: Immagine
2205         nullable: true
2206     type: object
2207     required:
2208     - id
2209     - title
2210     - creationDate
2211     - categoryId
2212 SavedInitiative:
2213     properties:
2214         userId:
2215             description: ID dell'utente che ha salvato l'iniziativa.
2216             type: integer
2217             minimum: -2147483648
2218             maximum: 2147483647
2219         initiativeId:
2220             description: ID dell'iniziativa salvata.
2221             type: integer
2222             minimum: -2147483648
2223             maximum: 2147483647
2224         savedAt:
2225             description: Data di salvataggio.
2226             default: current_timestamp()
2227             type: string
2228     type: object
2229     required:
2230     - userId
2231     - initiativeId
2232     - savedAt
2233 Notification:
2234     properties:
2235         id:
2236             description: ID univoco della notifica.
2237             type: integer
2238             minimum: 1
2239             maximum: 2147483647
2240         text:
2241             description: Contenuto della notifica.
2242             type: string
2243             maxLength: 65535
2244         isRead:
2245             type: boolean
2246             description: Flag che indica se è stata letta o no.
2247             nullable: true
2248         creationDate:
2249             description: Data di creazione.
2250             default: current_timestamp()
2251             type: string
2252         linkRef:
2253             description: ''
2254             default: 'NULL'
2255             type: string
2256             maxLength: 255

```

```

2257         nullable: true
2258     type: object
2259     required:
2260     - id
2261     - text
2262     - creationDate
2263 PBOption:
2264     properties:
2265         id:
2266             description: ID univoco dell'opzione.
2267             type: integer
2268             minimum: 1
2269             maximum: 2147483647
2270         text:
2271             description: Testo di descrizione dell'opzione.
2272             type: string
2273             maxLength: 250
2274         position:
2275             description: >-
2276                 Posizione relativa alla numerazione nell'elenco di opzioni del
2277                 bilancio.
2278             type: integer
2279             minimum: -2147483648
2280             maximum: 2147483647
2281     type: object
2282     required:
2283     - id
2284     - text
2285     - position
2286 Platform:
2287     properties:
2288         id:
2289             description: ID univoco della piattaforma.
2290             type: integer
2291             minimum: 1
2292             maximum: 2147483647
2293         platformName:
2294             description: Nome della piattaforma.
2295             type: string
2296             maxLength: 100
2297         iconPath:
2298             description: Path dell'icona.
2299             default: 'NULL'
2300             type: string
2301             maxLength: 255
2302             nullable: true
2303         platformLink:
2304             description: Link di reindirizzamento alla piattaforma.
2305             default: 'NULL'
2306             type: string
2307             maxLength: 255
2308             nullable: true
2309     type: object

```

```

2310     required:
2311       - id
2312       - platformName
2313   Reply:
2314     properties:
2315       id:
2316         description: ID univoco della risposta.
2317         type: integer
2318         minimum: 1
2319         maximum: 2147483647
2320     initiativeId:
2321       description: ID univoco dell'iniziativa a cui è rivolta la risposta.
2322       type: integer
2323       minimum: -2147483648
2324       maximum: 2147483647
2325     adminId:
2326       description: ID univoco dell'amministratore che ha fornito la risposta.
2327       default: 'NULL'
2328       type: integer
2329       minimum: -2147483648
2330       maximum: 2147483647
2331       nullable: true
2332     replyText:
2333       description: Testo di esposizione delle motivazioni della risposta.
2334       type: string
2335       maxLength: 65535
2336     creationDate:
2337       description: Data di creazione.
2338       default: current_timestamp()
2339       type: string
2340     attachments:
2341       type: array
2342       items:
2343         $ref: '#/components/schemas/AttachmentNoID'
2344       description: Eventuali allegati (immagini o PDF)
2345       nullable: true
2346   type: object
2347   required:
2348     - id
2349     - initiativeId
2350     - replyText
2351     - creationDate
2352   User:
2353     properties:
2354       id:
2355         description: ID univoco dell'utente.
2356         type: integer
2357         minimum: 1
2358         maximum: 2147483647
2359     firstName:
2360       description: Nome dell'utente.
2361       type: string
2362       maxLength: 100

```



```

2363     lastName:
2364         description: Cognome dell'utente.
2365         type: string
2366         maxLength: 100
2367     fiscalCode:
2368         description: Codice fiscale dell'utente.
2369         type: string
2370         maxLength: 16
2371     email:
2372         description: Email dell'utente.
2373         type: string
2374         maxLength: 150
2375     isAdmin:
2376         type: boolean
2377         description: >-
2378             Flag che determina se l'utente può svolgere il ruolo di
2379             amministratore.
2380     isCitizen:
2381         type: boolean
2382         description: Flag che determina se l'utente può svolgere il ruolo di cittadino.
2383     createdAt:
2384         description: Data di registrazione.
2385         default: current_timestamp()
2386         type: string
2387 type: object
2388 required:
2389     - id
2390     - firstName
2391     - lastName
2392     - fiscalCode
2393     - email
2394     - createdAt
2395     - isAdmin
2396     - isCitizen
2397     responses: {}
2398     securitySchemes: {}
2399 servers: []
2400 security: []

```

3 Implementazione

L'applicazione è stata sviluppata adottando un'architettura client-server disaccoppiata, basata interamente sull'ecosistema JavaScript.

Per la parte di Backend, è stato utilizzato il runtime environment Node.js con il framework Express.js per la creazione delle API RESTful. Questa scelta tecnologica è stata dettata dalla necessità di gestire un elevato numero di operazioni di I/O asincrone (come upload di file e invio email) in modo non bloccante.

Per la persistenza dei dati è stato adottato MySQL. La scelta di un RDBMS risponde alla natura intrinsecamente strutturata e stabile del dominio applicativo (gestione istituzionale), rendendo non necessaria la flessibilità schema-less delle soluzioni NoSQL. Questa architettura garantisce la rigorosa integrità referenziale e la consistenza transazionale (ACID) indispensabili per la validità di firme e votazioni. L'interazione con il database è gestita dalla libreria mysql2, che assicura efficienza e protezione contro SQL Injection tramite l'uso di prepared statements.

Per la parte di Frontend, lo sviluppo è stato realizzato con il framework Vue.js 3 (Composition API). È stato scelto Vite come strumento di build per la sua velocità di compilazione e l'Hot Module Replacement (HMR) istantaneo. La gestione dello stato applicativo è centralizzata tramite Pinia, mentre l'interfaccia utente è stata costruita seguendo un approccio a componenti modulari.

3.1 Organizzazione delle repositories

```
/
├── .github/ ..... Configurazioni specifiche di GitHub
│   ├── workflows/ ..... Automazioni e pipeline (GitHub Actions)
│   │   └── cicd.yml ..... Pipeline CI/CD per test e deploy automatico su Render
├── client/ ..... Frontend Application (Vue.js + Vite)
│   ├── src/
│   │   ├── assets/ ..... Immagini e stili compilati
│   │   ├── components/ ..... Componenti UI riutilizzabili (Cards, Toasts)
│   │   ├── composables/ ..... Logica riutilizzabile (Hooks custom)
│   │   ├── router/ ..... Configurazione delle rotte Vue Router
│   │   ├── stores/ ..... State Management (Pinia stores)
│   │   ├── utils/ ..... Funzioni di utilità (es. dateUtils)
│   │   ├── views/ ..... Pagine principali (Admin e User views)
│   │   ├── App.vue ..... Componente Root
│   │   └── main.js ..... Entry point frontend
│   ├── vite.config.js ..... Configurazione bundler Vite
│   └── package.json ..... Dipendenze Frontend
├── deliverable/ ..... Documentazione
│   └── D*/ ..... Deliverable numero * (main .tex, pdf, sections e immagini)
└── server/ ..... Backend Application
```

APIsDocumentazione/	Specifiche API (OpenAPI/Swagger)
DatabaseRelazionale/	Contiene lo script SQL per la creazione del DB
backend/	Server Node.js
_ __tests__/_	Test di integrazione e unità
_ data/_	Dati statici esterni
_ src/_	
_ config/_	Configurazione DB e costanti
_ controllers/_	Logica di business e gestione richieste
_ middleware/_	Auth, upload e gestione errori
_ routes/_	Definizione endpoint API
_ services/_	Logica complessa (es. importService)
_ utils/_	Utility backend
_ validators/_	Validazione schemi dati
_ app.js	Configurazione applicazione Express
_ server.js	Entry point server
_ uploads/_	File caricati dagli utenti
_ initiatives/_	Allegati iniziative
_ replies/_	Allegati risposte
_ package.json	Dipendenze Backend
_ README.md	Documentazione generale

3.2 Branching strategy e organizzazione del lavoro

Per la gestione del versionamento del codice è stata utilizzata la piattaforma GitHub. Data la dimensione del team e l'interdipendenza tra i moduli backend e frontend, abbiamo optato per una strategia di Trunk-Based Development.

A differenza di strategie complesse come GitFlow (spesso sovrabbondanti per team ristretti), lo sviluppo si è concentrato su un unico ramo principale, il branch main. Questa scelta ha garantito:

- **Integrazione Continua:** Ogni funzionalità completata è stata immediatamente integrata nel codice base, evitando il "merge hell" tipico dei branch a lunga vita.
- **Risoluzione Rapida dei Conflitti:** Lavorando sullo stesso ramo, eventuali conflitti tra il lavoro dei membri (es. modifiche concorrenti allo stesso file) venivano evidenziati e risolti immediatamente.

3.2.1 Suddivisione Logica del Lavoro

Sebbene il lavoro sia confluito tecnicamente sullo stesso branch, lo sviluppo è stato logicamente parallelizzato assegnando a ciascun membro la responsabilità esclusiva di specifiche aree funzionali, riducendo al minimo le sovrapposizioni:

Enea D'Angiò : Si è occupato principalmente dell'architettura Backend (configurazione Express, gestione e creazione del database MySQL) e della progettazione e integrazione

ne delle API. Ha contribuito parzialmente al Frontend attraverso bugfix, refactoring e implementazioni di piccole feature. Ha inoltre lavorato al deployment.

Ivan Nedeljkovic : Ha curato lo sviluppo dei componenti Frontend in Vue.js creando l'intera infrastruttura lato client e gestendo ui e ux. Ha inoltre collaborato al deployment.

Alessandro Mattarolo : Ha seguito lo sviluppo con la scrittura della documentazione, l'organizzazione dei file LaTeX e la progettazione del Frontend.

L'analisi dei contributi (visibile nella sezione Insights > Contributors della repository) mostra un totale di 102 commit, distribuiti tra i membri in modo proporzionale ai task assegnati. Lo sbilanciamento fra la quantità di commit eseguiti da Enea D'Angiò e Ivan Nedeljkovic e quelli eseguiti da Alessandro Mattarolo è dovuto al fatto che quest'ultimo ha lavorato principalmente alla documentazione.

3.3 Dipendenze

Il progetto si basa su un insieme di librerie gestite tramite il package manager **npm**. Di seguito sono elencate le dipendenze utilizzate, suddivise per ambito applicativo.

3.3.1 Backend

Le seguenti librerie sono state utilizzate per lo sviluppo del server Node.js:

cors Middleware per abilitare il Cross-Origin Resource Sharing, permettendo al frontend di comunicare con il backend.

dotenv Modulo per caricare le variabili d'ambiente da un file **.env** a **process.env**, garantendo la sicurezza delle credenziali.

express Framework web minimalista per Node.js, utilizzato per la gestione del server, del routing e delle API REST.

google-auth-library Libreria client di Google per la verifica sicura dei token di autenticazione e l'integrazione del login Google.

joi Libreria per la validazione degli schemi di dati, utilizzata per verificare la correttezza degli input nelle richieste API.

multer Middleware per la gestione di **multipart/form-data**, utilizzato per l'upload di file (immagini delle iniziative).

mysql2 Client MySQL veloce e sicuro per Node.js, supporta Promise e Prepared Statements per prevenire SQL Injection.

node-cron Scheduler per l'esecuzione di task periodici sul server (cron jobs).

nodemailer Modulo per l'invio di email transazionali (es. notifiche) tramite server SMTP.

nodemon Strumento di sviluppo che monitora le modifiche ai file e riavvia automaticamente il server.

3.3.2 Frontend

Per lo sviluppo dell'interfaccia utente con Vue.js sono state utilizzate le seguenti dipendenze:

vue Framework progressivo per la creazione di interfacce utente reattive e basate su componenti.

vue-router Router ufficiale per Vue.js, gestisce la navigazione SPA (Single Page Application).

pinia Store manager per Vue, utilizzato per la gestione centralizzata dello stato dell'applicazione.

axios Client HTTP basato su Promise per effettuare richieste asincrone verso il backend.

vite Build tool di nuova generazione che offre un ambiente di sviluppo rapido e ottimizzazione per la produzione.

vue3-google-login Componente Vue per facilitare l'integrazione del pulsante di accesso Google.

eslint / eslint-plugin-vue Strumenti di analisi statica per identificare problemi nel codice JavaScript e Vue.

prettier Code formatter per garantire uno stile di codice coerente.

@vitejs/plugin-vue Plugin ufficiale per il supporto di Vue.js all'interno di Vite.

vite-plugin-vue-devtools Plugin per migliorare l'esperienza di debugging con i devtools di Vue.

3.4 Database

Il database MySQL è stato strutturato nelle seguenti tabelle principali:

3.4.1 Categoria

La tabella **CATEGORIA** gestisce la classificazione tematica delle iniziative presenti sulla piattaforma.

- **ID_CATEGORIA** (INT): Chiave primaria autoincrementale che identifica univocamente la categoria.
- **NOME** (VARCHAR): Nome della categoria (es. "Ambiente", "Viabilità"), univoco nel sistema.

3.4.2 Piattaforma

La tabella PIATTAFORMA memorizza le informazioni relative ai siti esterni (es. Change.org) da cui possono essere importate le iniziative.

- **ID_PIATTAFORMA** (INT): Chiave primaria autoincrementale per identificare la piattaforma esterna.
- **NOME_PIATTAFORMA** (VARCHAR): Nome della piattaforma esterna.
- **PATH_ICONA** (VARCHAR): Percorso del file immagine dell'icona associata alla piattaforma.
- **LINK_BASE_PIATTAFORMA** (VARCHAR): URL base della piattaforma esterna.

3.4.3 Utente

La tabella UTENTE centralizza le informazioni di tutti gli attori che interagiscono con la piattaforma (Cittadini e Amministratori).

- **ID_UTENTE** (INT): Chiave primaria autoincrementale che identifica univocamente l'utente.
- **NOME** (VARCHAR): Nome anagrafico dell'utente.
- **COGNOME** (VARCHAR): Cognome anagrafico dell'utente.
- **CODICE_FISCALE** (CHAR): Codice fiscale univoco (formato standard 16 caratteri).
- **EMAIL** (VARCHAR): Indirizzo email univoco, utilizzato per l'accesso e le notifiche.
- **IS_ADMIN** (BOOLEAN): Definisce se l'utente possiede i privilegi di amministratore.
- **IS_CITTADINO** (BOOLEAN): Definisce se l'utente è verificato come residente/cittadino.
- **CREATED_AT** (TIMESTAMP): Data e ora di registrazione dell'utente.

3.4.4 Pre_Autorizzato

La tabella PRE_AUTORIZZATO funge da whitelist contenente i codici fiscali degli amministratori non cittadini e che quindi otterranno un account con privilegi da admin ma non da cittadino.

- **CODICE_FISCALE** (CHAR): Chiave primaria, contiene il codice fiscale pre-validato.
- **DATA_INSERIMENTO** (TIMESTAMP): Data in cui il codice è stato inserito nel sistema.
- **INSERITO_DA** (INT): Chiave esterna verso UTENTE, indica l'amministratore che ha inserito il dato.

3.4.5 Notifica

La tabella NOTIFICA gestisce i messaggi e gli avvisi inviati dal sistema agli utenti.

- **ID_NOTIFICA** (INT): Chiave primaria autoincrementale della notifica.
- **ID_UTENTE** (INT): Chiave esterna che associa la notifica al destinatario.
- **TESTO** (TEXT): Contenuto testuale del messaggio.
- **LETTA** (BOOLEAN): Flag che indica se la notifica è stata visualizzata dall'utente.
- **DATA_CREAZIONE** (TIMESTAMP): Data e ora di generazione della notifica.
- **LINK_RIF** (VARCHAR): Collegamento opzionale per reindirizzare l'utente all'evento specifico.

3.4.6 Bilancio Partecipativo

La tabella BILANCIO_PARTECIPATIVO definisce le consultazioni pubbliche create dall'amministrazione per l'allocazione di risorse.

- **ID_BIL** (INT): Chiave primaria autoincrementale dell'evento di bilancio.
- **ID_CREATOR** (INT): Chiave esterna verso l'amministratore che ha creato l'evento.
- **TITOLO** (VARCHAR): Titolo descrittivo del bilancio partecipativo.
- **CREATED_AT** (TIMESTAMP): Data di creazione dell'evento.
- **DATA_SCADENZA** (DATE): Data termine oltre la quale non è più possibile votare.

3.4.7 Iniziativa

La tabella INIZIATIVA costituisce il nucleo centrale dell'applicativo web ed include iniziative sia interne che esterne.

- **ID_INIZIATIVA** (INT): Chiave primaria autoincrementale dell'iniziativa.
- **TITOLO** (VARCHAR): Titolo dell'iniziativa.
- **DESCRIZIONE** (TEXT): Descrizione dettagliata della proposta.
- **LUOGO** (VARCHAR): Indicazione geografica o zona di interesse.
- **STATO** (ENUM): Stato corrente ('In corso', 'Approvata', 'Respinta', 'Scaduta').
- **NUM_FIRME** (INT): Contatore delle firme raccolte (aggiornato per iniziative interne o importato per esterne).

- **DATA_CREAZIONE** (TIMESTAMP): Data di pubblicazione dell'iniziativa.
- **DATA_SCADENZA** (DATE): Termine ultimo per la raccolta firme.
- **ID_AUTORE** (INT): Chiave esterna verso l'utente proponente (NULL se di sistema o importata anonimamente).
- **ID_CATEGORIA** (INT): Chiave esterna per la categorizzazione tematica.
- **ID_PIATTAFORMA** (INT): Chiave esterna valorizzata solo se l'iniziativa proviene da una piattaforma terza.
- **URL_ESTERNO** (VARCHAR): Link originale dell'iniziativa se esterna, NULL altrimenti.

3.4.8 Opzioni Bilancio

La tabella **OPZIONI_BILANCIO** contiene le singole voci o progetti votabili all'interno di un evento di bilancio partecipativo.

- **ID_OB** (INT): Chiave primaria autoincrementale dell'opzione.
- **ID_BIL** (INT): Chiave esterna che collega l'opzione al relativo bilancio partecipativo.
- **TEXT** (VARCHAR): Descrizione testuale dell'opzione.
- **POSITION** (TINYINT): Ordine numerico di visualizzazione nella lista.

3.4.9 Voti Bilancio

La tabella **VOTI_BILANCIO** registra le preferenze espresse dagli utenti per le opzioni di bilancio.

- **ID_UTENTE** (INT): Chiave esterna dell'utente votante (parte della chiave primaria composta).
- **ID_BIL** (INT): Chiave esterna del bilancio (parte della chiave primaria composta per garantire un solo voto per bilancio).
- **OPTION_ID** (INT): Chiave esterna dell'opzione scelta.
- **VOTED_AT** (TIMESTAMP): Data e ora dell'espressione del voto.

3.4.10 Firma Iniziativa

La tabella **FIRMA_INIZIATIVA** memorizza le adesioni degli utenti alle iniziative proposte.

- **ID_UTENTE** (INT): Chiave esterna dell'utente firmatario.
- **ID_INIZIATIVA** (INT): Chiave esterna dell'iniziativa firmata.
- **DATA_FIRMA** (TIMESTAMP): Data e ora della firma.

3.4.11 Iniziativa Salvata

La tabella INIZIATIVA_SALVATA permette agli utenti di salvare iniziative nei propri preferiti per consultazione futura.

- **ID_UTENTE** (INT): Chiave esterna dell'utente che salva l'iniziativa.
- **ID_INIZIATIVA** (INT): Chiave esterna dell'iniziativa salvata.
- **SAVED_AT** (TIMESTAMP): Data e ora del salvataggio.

3.4.12 Risposta

La tabella RISPOSTA contiene le comunicazioni ufficiali dell'amministrazione in merito a una specifica iniziativa.

- **ID_RISPOSTA** (INT): Chiave primaria autoincrementale della risposta.
- **ID_INIZIATIVA** (INT): Chiave esterna dell'iniziativa a cui la risposta si riferisce.
- **ID_ADMIN** (INT): Chiave esterna dell'amministratore che ha redatto la risposta.
- **TEXT_RISP** (TEXT): Contenuto della risposta ufficiale.
- **DATA_CREAZIONE** (TIMESTAMP): Data di pubblicazione della risposta.

3.4.13 Allegato

La tabella ALLEGATO gestisce i file multimediali associati alle entità del sistema, implementando un vincolo di esclusività (XOR) tra Iniziative e Risposte.

- **ID_ALLEGATO** (INT): Chiave primaria autoincrementale del file.
- **FILE_NAME** (VARCHAR): Nome originale del file.
- **FILE_PATH** (VARCHAR): Percorso di archiviazione del file nel server.
- **FILE_TYPE** (VARCHAR): Tipologia del file (estensione).
- **UPLOADED_AT** (TIMESTAMP): Data di caricamento.
- **ID_INIZIATIVA** (INT): Chiave esterna (opzionale) se l'allegato appartiene a un'iniziativa.
- **ID_RISPOSTA** (INT): Chiave esterna (opzionale) se l'allegato appartiene a una risposta.

3.5 Testing

Le api e il relativo codice presentano una test-suite che consente di verificarne il corretto funzionamento. I test sono utilizzati all'interno della configurazione di CI/CD.

L'implementazione dei test è organizzata in file .test.js implementati nella cartella __test__. Sono divisi in 4 file, organizzati per argomento.

Abbiamo implementato tutti i test definiti, per un totale di 94 test.

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
1.1	Login con token Google non valido	token="INVALID"	Il client invia un token che Google non riesce a validare o che risulta malformato.	-	Err 401 "Token non valido"	Err 401 "Token non valido"	RF1.1
1.2	Login utente esistente (match Codice Fiscale)	token="VALID_EXISTING"	L'utente deve essere già registrato nel database e il Codice Fiscale calcolato dal token Google deve corrispondere esattamente a quello salvato.	DB	Status 200 JSON Utente	Status 200 JSON Utente	RF1.2
1.3	Login utente esistente (fallback Email)	token="VALID_EMAIL"	L'utente non viene trovato per Codice Fiscale, ma esiste un utente con la stessa email. Il CF nel DB è diverso da quello calcolato.	DB	Status 200 CF Aggiornato	Status 200 CF Aggiornato	RF1.3
1.4	Login nuovo utente (non registrato)	token="VALID_NEW"	L'utente non deve essere presente nel database né per CF né per email, e non deve essere nella lista dei pre-autorizzati.	DB	Err 404 NEED_REGISTRATION	Err 404 NEED_REGISTRATION	RF1.4
1.5	Login Admin pre-autorizzato	token="PREAUTH_ADMIN"	L'utente non esiste in anagrafica, ma il suo CF è presente nella tabella PRE_AUTORIZZATO inserita da un altro admin.	DB	Status 200 Profilo Admin creato	Status 200 Profilo Admin creato	RF2.2
2.1	Richiesta OTP: Email non valida	email="not-an-email"	Il formato dell'email inviata non rispetta la regex standard per gli indirizzi email.	-	Err 400 "Email non valida"	Err 400 "Email non valida"	RF2.0
2.2	Richiesta OTP: Email già usata	email="existent@test.com"	L'indirizzo email specificato è già associato a un utente attivo nel database UTENTE.	DB	Err 409 "Email già utilizzata"	Err 409 "Email già utilizzata"	RF2.0

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
2.3	Richiesta OTP: Successo	email="new@t est.co m"	L'email è valida e non è presente nel database. Il servizio di posta (o mock) deve essere pronto.	SMTP	Status 200 "Codice inviato"	Status 200 "Codice inviato"	RF2.0
2.4	Richiesta OTP: DevMode	email="dev@t est.co m"	Le credenziali SMTP non sono configurate nell'ambiente (variabili d'ambiente mancanti).	Env	Status 200 "Check conso- le"	Status 200 "Check conso- le"	Dev
3.1	Registra- zione completa- mento Cittadino	otp="1 23456", token= "VALID"	È stato generato un OTP valido per l'email associata al token Google e l'OTP non è ancora scaduto.	Redis / Mem	Status 201 Utente creato	Status 201 Utente creato	RF2.1
3.2	Registra- zione Admin (via email)	otp="6 54321", email= "...ad min..."	L'email contiene la sottostringa "admin" (logica di test) e l'OTP è valido.	Redis / Mem	Status 201 Admin creato	Status 201 Admin creato	RF2.1
3.3	Registra- zione: OTP Errato	otp="9 99999"	L'OTP fornito non corrisponde a quello salvato temporaneamente per l'email specificata.	Redis / Mem	Err 400 "Codice OTP errato"	Err 400 "Codice OTP errato"	RF2.0
3.4	Registra- zione: OTP Scaduto	otp="1 11111"	L'OTP salvato ha superato il tempo di validità (es. 5 minuti) ed è considerato scaduto.	Redis / Mem	Err 400 "Codice OTP scadu- to"	Err 400 "Codice OTP scadu- to"	RF2.0
3.5	Registra- zione: Token Google Invalido	google Token= "INVAL ID"	L'OTP è corretto ma il token Google fornito per la verifica finale non è valido o è scaduto.	Google API	Err 401 "Token non valido"	Err 401 "Token non valido"	RF2.0
3.6	Registra- zione: Email duplicata	email="dup@t est.co m"	Si verifica una race condition dove l'email viene inserita nel DB tra la richiesta OTP e la conferma registrazione.	DB	Err 409 "Email già regi- strata"	Err 409 "Email già regi- strata"	RF2.0

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
4.0	Logout	-	Nessuna precondizione particolare (stateless), la richiesta deve essere processata correttamente.	-	Status 200 "Logout effettuato"	Status 200 "Logout effettuato"	RF5
5.1	Creazione iniziativa (Successo)	title, desc, place, catId, file	L'utente deve essere autenticato come Cittadino. Tutti i campi obbligatori sono presenti e validi.	DB	Status 201 ID Iniziativa	Status 201 ID Iniziativa	RF1
5.2	Creazione iniziativa: Validazione	title (manca descrizione)	L'utente è autenticato. Il payload manca di uno o più campi obbligatori (es. descrizione).	-	Err 400 Bad Request	Err 400 Bad Request	-
5.3	Creazione iniziativa: Non autenticato	title, desc	L'intestazione della richiesta non contiene un token di sessione valido o nessun utente è loggato.	-	Err 401 Unauthorized	Err 401 Unauthorized	-
5.4	Creazione iniziativa: Cooldown	title, desc	L'utente ha già creato un'iniziativa ed è trascorsa meno di una soglia di tempo (14 giorni) da tale creazione.	DB	Err 422 "Già creato di recente"	Err 422 "Già creato di recente"	RF13
6.1	Lista iniziative: Paginazione	page=1, limit=2	Devono esistere nel database più iniziative di quante richieste per pagina per verificare lo split.	DB	Status 200 Array limitato	Status 200 Array limitato	-
6.2	Lista iniziative: Filtro Stato	status="Approvata"	Il database deve contenere iniziative con stati diversi (Approvata, In Corso, Respinta).	DB	Status 200 Solo "Approvata"	Status 200 Solo "Approvata"	RF3
6.3	Lista iniziative: Filtro Categoria	category=3	Il database deve contenere iniziative associate a diverse categorie.	DB	Status 200 Solo Cat. 3	Status 200 Solo Cat. 3	RF3
6.4	Lista iniziative: Filtro Firme	minSignatures=100	Devono esistere iniziative con un numero di firme superiore e inferiore alla soglia.	DB	Status 200 Firme >= 100	Status 200 Firme >= 100	RF3

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
6.5	Lista iniziative: Ricerca testo	<code>search="Libro"</code>	Deve esistere almeno un'iniziativa il cui titolo contenga la parola chiave specificata.	DB	Status 200 Include "Libro"	Status 200 Include "Libro"	-
6.6	Lista iniziative: Ordinamento	<code>sortBy=1, order="asc"</code>	Le iniziative devono avere date di creazione diverse per verificare l'ordinamento temporale.	DB	Status 200 Ordine cronologico	Status 200 Ordine cronologico	-
7.1	Dettaglio Iniziativa: Successo	<code>id=101</code>	L'ID richiesto deve corrispondere a un'iniziativa esistente nel database.	DB	Status 200 Oggetto Iniziativa	Status 200 Oggetto Iniziativa	-
7.2	Dettaglio Iniziativa: Non Trovato	<code>id=9999</code>	L'ID richiesto non deve corrispondere ad alcuna iniziativa presente nel database.	DB	Err 404 Not Found	Err 404 Not Found	-
8.1	Patch Iniziativa: Admin Estende Data	<code>expirationDate="2099..."</code>	L'utente deve essere Admin. L'iniziativa deve esistere. La nuova data deve essere futura.	DB	Status 200 Data aggiornata	Status 200 Data aggiornata	RF7
8.2	Patch Iniziativa: Cittadino non autorizzato	<code>expirationDate="2099..."</code>	L'utente è autenticato ma ha ruolo di Cittadino (non Admin).	-	Err 403 Forbidden	Err 403 Forbidden	RF7
9.1	Risposta Admin: Approvazione	<code>status="Approvata", motivo</code>	L'utente deve essere Admin. L'iniziativa esiste ed è in attesa di valutazione.	DB	Status 201 Stato cambiato	Status 201 Stato cambiato	RF7
9.2	Risposta Admin: Cittadino vietato	<code>status="Approvata"</code>	L'utente è un Cittadino e tenta di chiamare l'endpoint riservato alla moderazione.	-	Err 403 Forbidden	Err 403 Forbidden	RF7
9.3	Risposta Admin: Stato Invalido	<code>status="Invalido"</code>	L'utente è Admin ma fornisce una stringa di stato non prevista dall'enum del sistema.	-	Err 400 Bad Request	Err 400 Bad Request	-

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
10.1	Firma iniziativa: Successo	id=103	L'utente è Cittadino e non ha ancora apposto la firma digitale a questa specifica iniziativa.	DB	Status 201 "Firma registrata"	Status 201 "Firma registrata"	RF11
10.2	Firma iniziativa: Doppia firma	id=103	L'utente ha già firmato l'iniziativa in precedenza (record presente in FIRMA_INIZIATIVA).	DB	Err 409 "Già firmato"	Err 409 "Già firmato"	RF11
10.3	Firma iniziativa: Anonimo	id=103	Nessun utente è loggato nella sessione corrente.	-	Err 401 Unauthorized	Err 401 Unauthorized	RF11
11.1	Follow iniziativa: Successo	id=104	L'utente è autenticato e non sta seguendo l'iniziativa (non presente in INIZIATIVA_SALVATA).	DB	Status 200 "Aggiunta preferiti"	Status 200 "Aggiunta preferiti"	RF14
11.2	Follow iniziativa: Idempotenza	id=104	L'utente sta già seguendo l'iniziativa. Il sistema non deve generare errore ma confermare.	DB	Status 200 Success	Status 200 Success	-
11.3	Unfollow iniziativa	id=104	L'utente sta seguendo l'iniziativa e richiede la rimozione dai preferiti.	DB	Status 200 "Rimos- sa seguiti"	Status 200 "Rimos- sa seguiti"	RF14
12.1	Crea Bilancio: Successo Admin	title, options[], expDate	L'utente è Admin. Non ci sono altri bilanci attivi. Le opzioni sono tra 2 e 5.	DB	Status 201 ID Bilancio	Status 201 ID Bilancio	RF8
12.2	Crea Bilancio: Cittadino	title, options[]	L'utente è un Cittadino e non ha i permessi per creare un bilancio partecipativo.	-	Err 403 Forbidden	Err 403 Forbidden	-
12.3	Crea Bilancio: Durata insufficiente	expDate = oggi + 5gg	L'utente è Admin, ma la data di scadenza è inferiore al minimo richiesto (14 giorni).	-	Err 400/422 Durata < 14gg	Err 400/422 Durata < 14gg	-
12.4	Crea Bilancio: Opzioni insufficienti	options (lunghezza 1)	L'array delle opzioni contiene un solo elemento. Minimo richiesto è 2.	-	Err 400 Bad Request	Err 400 Bad Request	-

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
12.5	Crea Bilancio: Troppe Opzioni	options (lunghezza 6)	L'array delle opzioni contiene più di 5 elementi. Massimo consentito è 5.	-	Err 400 Bad Request	Err 400 Bad Request	-
12.6	Crea Bilancio: Conflitto Attivo	title, options[]	Esiste già un bilancio nel sistema con stato 'Attivo' (data scadenza futura).	DB	Err 409 "Bilancio attivo"	Err 409 "Bilancio attivo"	-
13.1	Consultazione Bilancio Attivo	-	Esiste un bilancio con data scadenza futura. Nessun filtro utente applicato.	DB	Status 200 Dati + Opzioni	Status 200 Dati + Opzioni	-
13.2	Consultazione con Voto Utente	Header X-Mock-User	L'utente specificato ha già votato per questo bilancio. L'API deve restituire l'ID dell'opzione votata.	DB	Status 200 votedOptionId	Status 200 votedOptionId	-
13.3	Consultazione: Nessun Attivo	-	Non ci sono bilanci attivi nel database al momento della richiesta.	DB	Status 200 Dati null	Status 200 Dati null	-
14.1	Votazione: Cittadino Successo	position=1	L'utente è Cittadino, il bilancio è attivo e l'utente non ha ancora votato.	DB	Status 200 Voto registrato	Status 200 Voto registrato	RF15
14.2	Votazione: Doppia (Conflitto)	position=2	L'utente ha già registrato un voto per questo stesso bilancio in precedenza.	DB	Err 409 "Già votato"	Err 409 "Già votato"	RF15
14.3	Votazione: Admin	position=1	L'utente è autenticato come Amministratore (non può votare nei bilanci partecipativi).	-	Err 403 Solo cittadini	Err 403 Solo cittadini	RF15
14.4	Votazione: Bilancio Scaduto	position=1	Il bilancio esiste ma la sua data di scadenza è passata.	DB	Err 403 Scaduto	Err 403 Scaduto	-
14.5	Votazione: Opzione Inesistente	position=99	L'ID o la posizione dell'opzione inviata non corrisponde a nessuna opzione del bilancio corrente.	DB	Err 400 Non esiste	Err 400 Non esiste	-
14.6	Votazione: Bilancio non trovato	id=9999	L'ID del bilancio passato nell'URL non esiste nel database.	DB	Err 404 Non trovato	Err 404 Non trovato	-

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
15.1	Archivio Storico: Lista Admin	page=1	L'utente è Admin. Esistono bilanci scaduti nel database.	DB	Status 200 Lista bilanci	Status 200 Lista bilanci	RF9
15.2	Archivio Storico: Paginazione	limit=2	Esistono più bilanci scaduti del limite per pagina impostato.	DB	Status 200 Limite rispettato	Status 200 Limite rispettato	-
15.3	Archivio Storico: Cittadino	-	L'utente è un Cittadino e tenta di accedere all'archivio storico (riservato admin).	-	Err 403 Forbidden	Err 403 Forbidden	-
15.4	Archivio Storico: Conteggio	-	I bilanci restituiti devono includere il campo dei voti totali per ogni opzione.	DB	Status 200 votes presente	Status 200 votes presente	-
16.1	Profilo Utente: Non Autenticato	-	Nessun token di sessione fornito nell'header della richiesta.	-	Err 401 Accesso negato	Err 401 Accesso negato	-
16.2	Profilo Utente: Successo	-	L'utente è autenticato. Il database contiene i dati anagrafici corretti per l'ID utente.	DB	Status 200 JSON Profilo	Status 200 JSON Profilo	RF12
16.3	Iniziativa Utente: Manca Relation	-	Richiesta GET senza il parametro query obbligatorio relation .	-	Err 400 Bad Request	Err 400 Bad Request	-
16.4	Iniziativa Utente: Create	relation=created	L'utente ha creato almeno un'iniziativa in passato.	DB	Status 200 Lista Create	Status 200 Lista Create	-
16.5	Iniziativa Utente: Firmate	relation=signed	L'utente ha firmato almeno un'iniziativa creata da altri.	DB	Status 200 Lista Firmate	Status 200 Lista Firmate	-
16.6	Iniziativa Utente: Seguite	relation=followed	L'utente segue almeno un'iniziativa (preferiti).	DB	Status 200 Lista Seguite	Status 200 Lista Seguite	-
16.7	Iniziativa Utente: Paginazione	page=1, limit=5	La lista risultante (create/firmate/seguite) contiene elementi paginabili.	DB	Status 200 Meta pagination	Status 200 Meta pagination	-

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
17.1	Notifiche: Lista	-	L'utente ha delle notifiche assegnate nel database.	DB	Status 200 Lista Notifiche	Status 200 Lista Notifiche	RF20
17.2	Notifiche: Isolamento Utenti	-	Esistono notifiche per altri utenti (es. Admin) nel DB. Queste non devono apparire.	DB	Status 200 Solo proprie	Status 200 Solo proprie	RF20
17.3	Notifiche: Filtro Non Lette	read=false	L'utente ha sia notifiche lette che non lette nel database.	DB	Status 200 Solo non lette	Status 200 Solo non lette	-
17.4	Notifiche: Segna come Letta	id=ID_NOTIF, isRead=true	La notifica esiste, appartiene all'utente ed è attualmente non letta.	DB	Status 200 DB Aggiornato	Status 200 DB Aggiornato	RF20
17.5	Notifiche: Body Invalido	isRead=false	Il payload della richiesta non è coerente con l'operazione (es. settare false o campo mancante).	-	Status 200/400	Status 200/400	-
17.6	Notifiche: Non Trovata	id=8888	L'ID della notifica non esiste nel database.	DB	Err 404 Not Found	Err 404 Not Found	-
17.7	Notifiche: Sicurezza Accesso	id=ID_ADMIN_NOTIF	La notifica esiste ma appartiene a un altro utente (es. Admin). L'utente corrente non deve poterla modificare.	DB	Err 404 (Security)	Err 404 (Security)	RF20
18.1	Gestione Admin: Lista (401)	isAdmin=true	Utente non autenticato tenta di accedere alla lista utenti.	-	Err 401 Unauthorized	Err 401 Unauthorized	-
18.2	Gestione Admin: Lista (403)	isAdmin=true	Utente Cittadino (autenticato) tenta di accedere alla lista admin.	-	Err 403 Forbidden	Err 403 Forbidden	-
18.3	Gestione Admin: Lista Successo	isAdmin=true	Utente Admin richiede la lista. Nel DB sono presenti almeno 2 admin.	DB	Status 200 Lista Admin	Status 200 Lista Admin	RF10

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
18.4	Gestione Admin: Filtro CF	fiscal Code="ADMIN2"	Esiste un utente admin con il codice fiscale parziale specificato.	DB	Status 200 Match CF	Status 200 Match CF	-
19.1	Modifica Ruolo: Non Auth	isAdmin=true	Utente non autenticato tenta di promuovere un utente.	-	Err 401 Unauthorized	Err 401 Unauthorized	-
19.2	Modifica Ruolo: Cittadino	isAdmin=true	Utente Cittadino tenta di promuovere un altro utente.	-	Err 403 Forbidden	Err 403 Forbidden	-
19.3	Modifica Ruolo: Promozione	isAdmin=true	Utente Admin modifica utente Cittadino. L'utente target esiste.	DB	Status 200 Ruolo Aggiornato	Status 200 Ruolo Aggiornato	RF10
19.4	Modifica Ruolo: Revoca	isAdmin=false	Utente Admin modifica un altro Admin per degradarlo a Cittadino.	DB	Status 200 Ruolo Aggiornato	Status 200 Ruolo Aggiornato	RF10
19.5	Modifica Ruolo: Self-Demotion	isAdmin=false	Utente Admin tenta di revocare i privilegi a se stesso (vietato per sicurezza).	-	Err 400/403	Err 400/403	-
19.6	Modifica Ruolo: Body Invalido	isAdmin="string"	Il valore passato per isAdmin non è un booleano valido.	-	Err 400 Bad Request	Err 400 Bad Request	-
19.7	Modifica Ruolo: Utente Inesistente	id=99999	L'ID utente target non esiste nel database.	DB	Err 404 Not Found	Err 404 Not Found	-
20.1	Ricerca Utente: Per CF	fiscal Code="CITIZEN1..."	L'utente Admin cerca un utente specifico esistente tramite CF esatto.	DB	Status 200 Utente Trovato	Status 200 Utente Trovato	-
20.2	Ricerca Utente: Parametri Mancanti	-	Richiesta GET senza parametri di ricerca obbligatori (es. fiscalCode o isAdmin).	-	Err 400 Bad Request	Err 400 Bad Request	-
20.3	Ricerca Utente: Non Trovato	fiscal Code="NONEXIST"	Nessun utente corrisponde al criterio di ricerca fornito.	DB	Err 404 Not Found	Err 404 Not Found	-

continua nella pagina successiva...

...continua dalla pagina precedente

N.	Desc.	Test Data	Precondizioni	Dip.	R. Atteso	R. Risc.	Note
21.1	Pre-autorizzazione: NonAdmin	fiscal Code="NEW..."	Utente Cittadino tenta di pre-autorizzare un nuovo amministratore.	-	Err 403 Forbidden	Err 403 Forbidden	-
21.2	Pre-autorizzazione: Successo	fiscal Code="PREAUTH..."	Utente Admin inserisce un nuovo CF non esistente né in UTENTE né in PRE_AUTORIZZATO.	DB	Status 201 Inserito	Status 201 Inserito	-
21.3	Pre-autorizzazione: Confitto	fiscal Code="EXISTING"	Il CF inserito appartiene già a un utente registrato o è già pre-autorizzato.	DB	Err 409 Conflict	Err 409 Conflict	-
22.1	Sicurezza: Accesso Risorse Altrui	id=NOT IF_ADMIN	Un utente Cittadino tenta di accedere tramite API diretta a una risorsa (es. notifica) di un Admin.	DB	Err 404 Not Found	Err 404 Not Found	-

4 FrontEnd

Di seguito viene illustrata la parte FrontEnd del progetto.

Premessa: le immagini riportate mostrano l'applicazione con il tema "dark", ma è disponibile anche una versione "light", ottenibile premendo l'icona del sole raffigurata nella parte più a destra dell'header. Quando si sta utilizzando la versione "light", l'icona raffigura una luna e premendola si ritorna in versione "dark".

4.1 Dettaglio dell'header

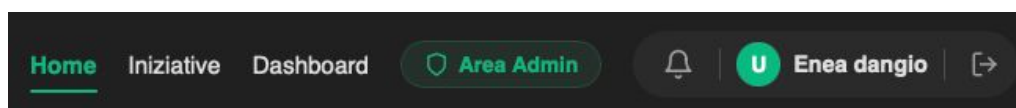


Figura 1: Header di un cittadino amministratore autenticato

Nella Figura 1 è riportato l'header presente nelle varie pagine dell'applicazione. Come già specificato nel Desing del FrontEnd (si veda il documento D1), il suo aspetto varia in funzione della tipologia d'utente. Nell'esempio riportato, ad esempio, assumiamo che l'utente sia autenticato e che abbia sia i privilegi da cittadino sia quelli da amministratore. Le altre varianti dell'header sono:

- Se l'utente non è autenticato, visualizza solo le sezioni "Home" e "Iniziative" nella barra in alto. Inoltre, al posto della campanella delle notifiche, dell'icona del suo profilo, del suo nome e del pulsante di logout visualizza un pulsante con la scritta "Accedi".
- Se l'utente autenticato non ha il ruolo di amministratore, non visualizza la sezione "Area Admin".
- Se l'utente non è un cittadino, non visualizza la sezione "Dashboard" e la campanella delle notifiche.

Cliccando su una delle sezioni, l'utente viene reindirizzato alla pagina dedicata. Se l'utente clicca sulla campanella visualizza le notifiche recenti.

4.2 Home Page

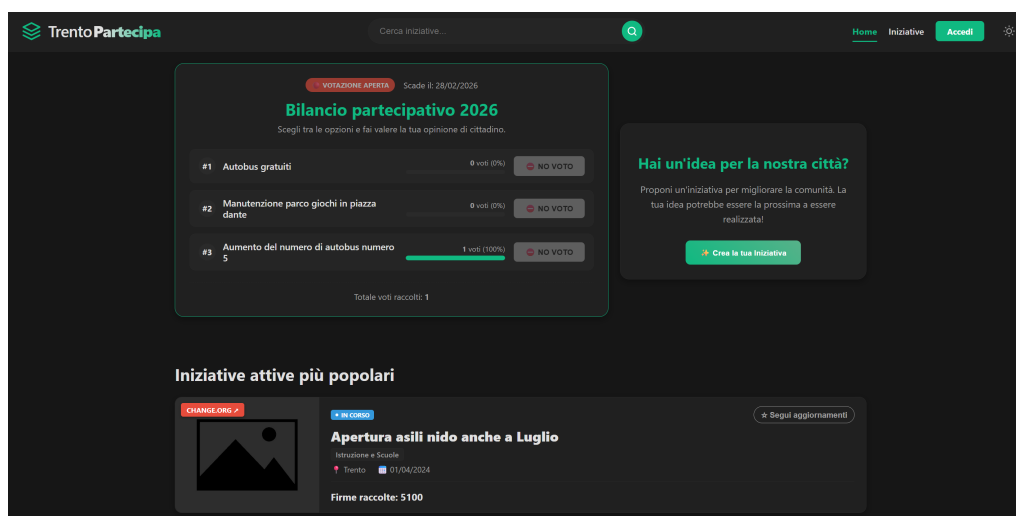


Figura 2: Home Page di Trento Partecipa

La Figura 2 raffigura la home page di "TRENTO PARTECIPA". Osserviamo che nell'header, oltre alla lista di sezioni in cui l'utente può navigare, c'è pure il logo e il nome dell'applicazione e una barra di ricerca il cui scopo è quello di permettere la ricerca di iniziative tramite parole chiave.

Nella home sono inoltre presenti il bilancio partecipativo attualmente in corso, il pulsante per creare una nuova iniziativa e una preview della lista di iniziative, ridotta alle iniziative in corso con più firme raccolte.

- **Bilancio partecipativo** - Come da specifiche, il bilancio partecipativo riporta la sua data di scadenza, le varie opzioni votabili, il numero di voti totale (in basso) e per opzione (a lato di ogni opzione), associati alla percentuale sul totale. I pulsanti per votare sono disabilitati per gli utenti che non sono cittadini autenticati, così come per i cittadini se hanno già votato.
- **Pulsante per creare un'iniziativa** - Cliccando il tasto, l'utente cittadino viene condotto alla schermata in Figura 5. Se un utente non autenticato clicca il pulsante, viene condotto alla schermata di login (Figura 3). Il pulsante è disabilitato per gli utenti autenticati senza i privilegi da cittadino.
- **Iniziative più popolari** - Rispecchiano le caratteristiche generali di tutte le iniziative, discusse successivamente in relazione alla Figura 6.

4.3 Schermata di login

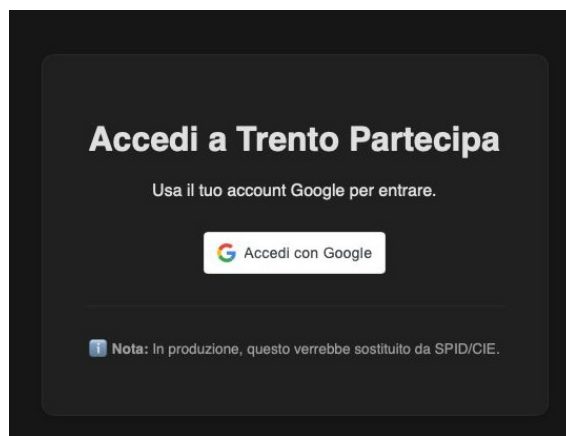


Figura 3: Schermata di login

Nella Figura 3 è raffigurata la schermata di login dell'applicazione, a cui l'utente viene condotto quando clicca sul tasto "Accedi" presente nell'header o, in alcuni casi, quando tenta di compiere delle azioni che possono essere compiute solo da utenti autenticati (ad esempio creare una nuova iniziativa). Date le complicanze introdotte dall'integrazione di SPID/CIE nel sistema, ci siamo limitati a implementare l'accesso tramite Google. Se l'utente sta eseguendo il suo primo accesso, viene condotto alla schermata di creazione di un nuovo profilo (Figura 4), altrimenti viene ricondotto alla home ma con i privilegi aggiornati in base al suo ruolo.

4.4 Schermata di creazione di un nuovo profilo

Quasi finito!

Conferma la tua email per ricevere il codice di sicurezza.

La tua Email

enea.dangio162006@gmail.com

Invia Codice OTP

Non divideremo mai la tua email con terze parti.

Verifica Sicurezza

Abbiamo inviato un codice a
enea.dangio162006@gmail.com

Scade tra: 4:55

Codice OTP a 6 cifre

[][][][][][]

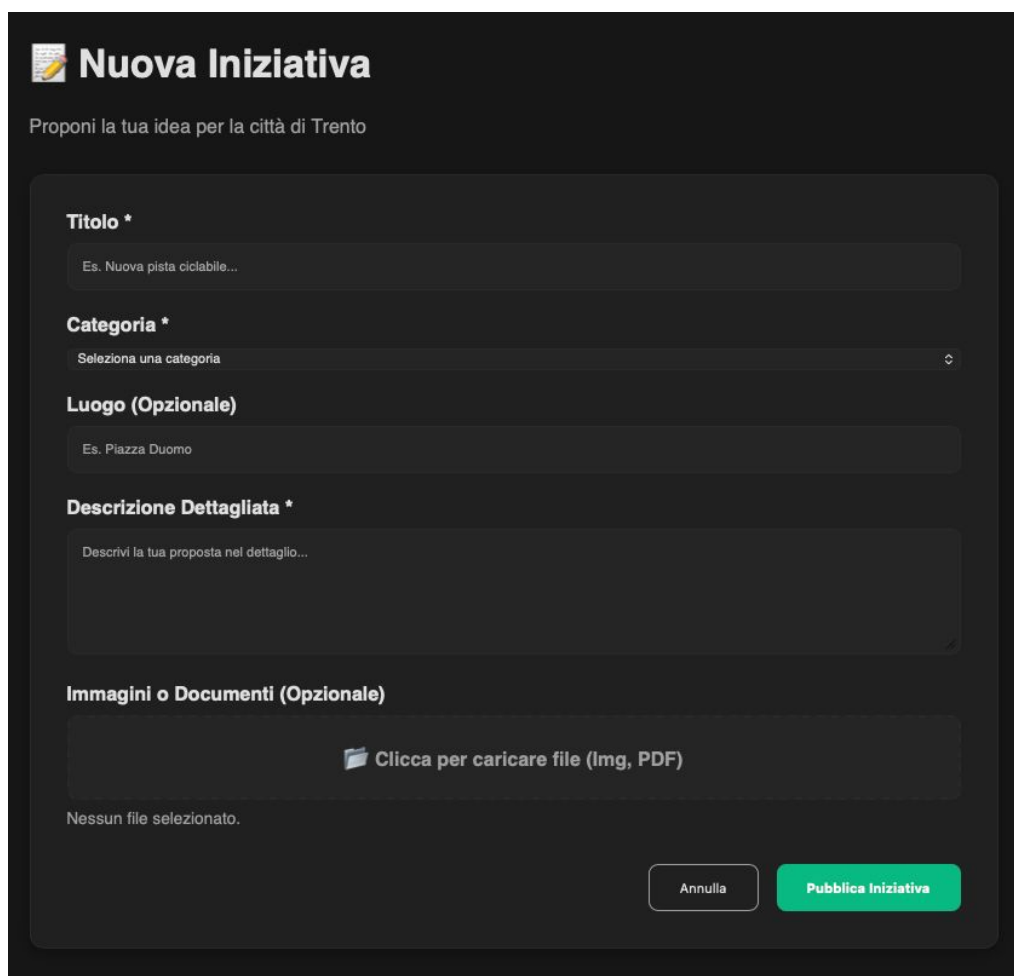
Conferma Registrazione

[Ho sbagliato email](#)

Figura 4: Le due fasi di creazione di un nuovo profilo

In Figura 4 sono raffigurate le due fasi di creazione di un nuovo profilo da parte di un utente. L'utente inserisce il proprio recapito email e attende poi di ricevere un codice OTP da inserire per la verifica di sicurezza. Nella prima fase sono presenti, chiaramente, il box per digitare l'email e il pulsante per l'invio del codice OTP. Nella seconda schermata sono presenti un timer che indica il tempo rimasto prima che il codice scada, i box per l'inserimento delle cifre del codice, il pulsante per confermare la propria registrazione e un pulsante "Ho sbagliato email" per tornare alla fase precedente.

4.5 Schermata di creazione di una nuova iniziativa



Nuova Iniziativa

Proponi la tua idea per la città di Trento

Titolo *

Es. Nuova pista ciclabile...

Categoria *

Seleziona una categoria

Luogo (Opzionale)

Es. Piazza Duomo

Descrizione Dettagliata *

Descrivi la tua proposta nel dettaglio...

Immagini o Documenti (Opzionale)

Clicca per caricare file (img, PDF)

Nessun file selezionato.

Annulla Pubblica Iniziativa

Figura 5: Schermata di creazione di una nuova iniziativa

In Figura 5 è raffigurata la schermata da cui l'utente può compilare i campi per creare e pubblicare una nuova iniziativa. Sono presenti i form per digitare il titolo, l'eventuale luogo e la descrizione, selezionare una categoria e allegare eventualmente delle immagini o dei documenti. In basso a destra sono raffigurati i pulsanti per annullare o confermare la pubblicazione.

4.6 Schermata di visualizzazione delle iniziative

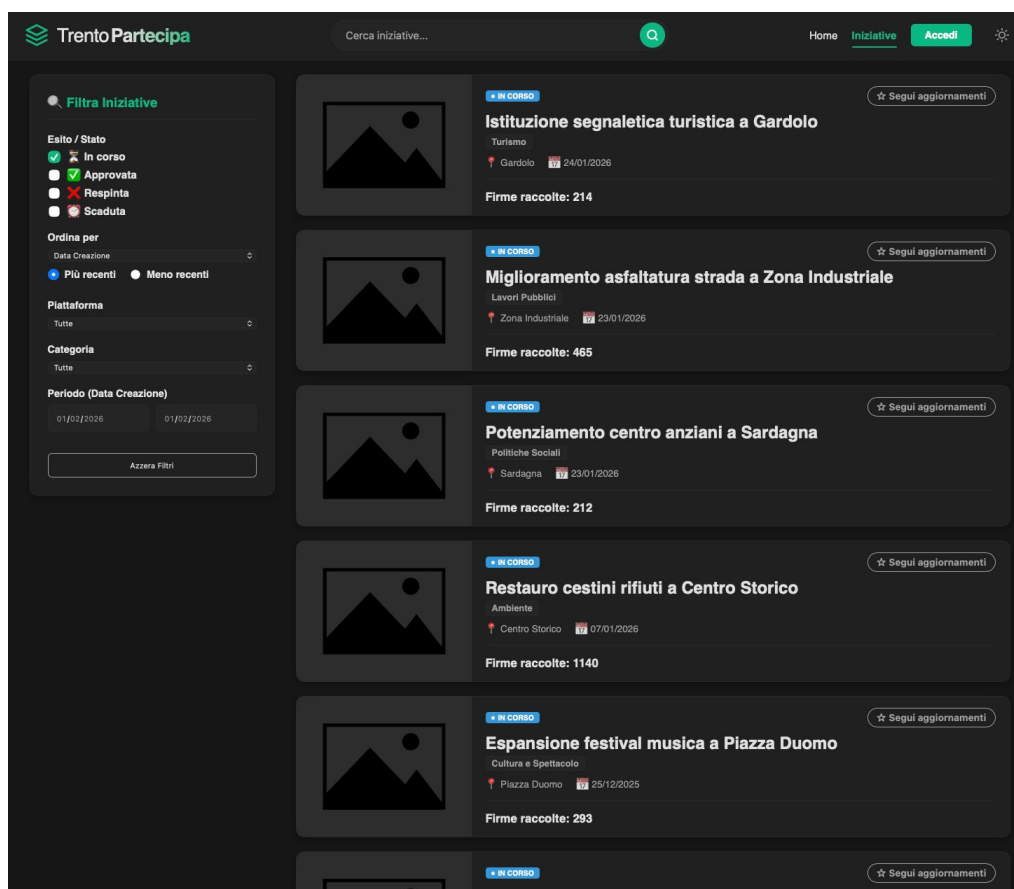


Figura 6: Schermata di visualizzazione delle iniziative

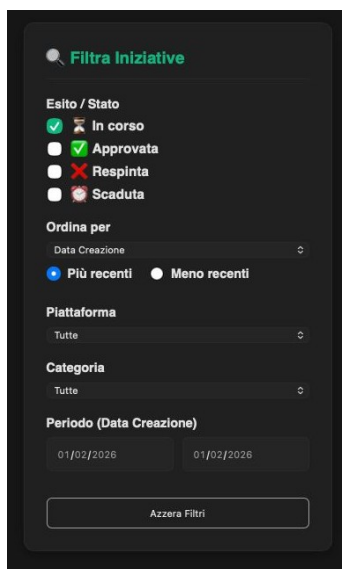
In Figura 6 è raffigurata la pagina sotto la voce "Iniziativa", accessibile da qualsiasi utente. In essa sono presenti le preview delle varie iniziative presenti in piattaforma e un box dove l'utente può applicare dei parametri di ricerca per visualizzare distintamente un certo sottoinsieme di iniziative. Se la quantità di iniziative da mostrare è maggiore di 10, queste vengono distribuite in diverse pagine numerate e navigabili, ognuna contenente al massimo 10 preview.

Dettaglio preview di un'iniziativa



Ogni preview di iniziativa riporta un titolo, la categoria, la data di pubblicazione, il numero di firme raccolte ed eventualmente un'immagine e il luogo. Se l'iniziativa è stata importata da una piattaforma esterna, questa viene indicata esplicitamente (si vedano ad esempio le preview presenti nella home in Figura 2). La preview presenta inoltre una label che ne indica lo stato (ad esempio "In corso") e un pulsante per seguirla, ossia aggiungerla alla dashboard personale. Se un utente non cittadino prova a seguire un'iniziativa, viene condotto alla schermata da cui fare il login (Figura 3). Cliccando sulla preview, si apre la pagina di dettaglio dell'iniziativa (Figura 7).

Dettaglio selezione dei criteri di ricerca



Oltre alla ricerca per parole chiave, possibile grazie alla barra di ricerca nell'header, l'utente può applicare anche dei criteri di ricerca tramite l'interfaccia in figura: è possibile filtrare le iniziative per stato, piattaforma, categoria, intervallo temporale (riferito alla data di creazione delle iniziative) e ordinarle in base alla data di creazione o al numero di firme. I filtri possono essere rimossi manualmente o con un apposito pulsante "Azzerare filtri".

4.7 Schermata di visualizzazione della singola iniziativa

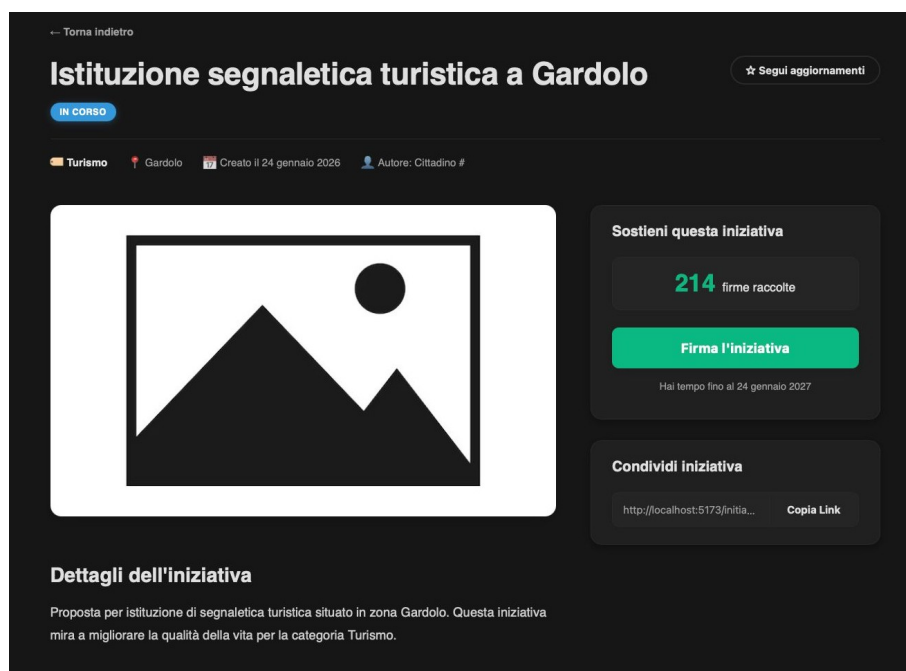


Figura 7: Schermata di visualizzazione di una singola iniziativa

In Figura 7 è raffigurata la pagina di dettaglio di un'iniziativa, che riporta titolo, stato, categoria, eventuale luogo, data di creazione e di scadenza, autore, descrizione, numero di firme, eventuali immagini ed allegati e, sotto la descrizione, la risposta del Comune, se presente (Figura 12). Sono poi presenti un tasto per copiare il link dell'iniziativa, in caso l'utente volesse condividerla, e il pulsante per firmarla (disabilitato se già premuto o se l'utente non è un cittadino). Se un utente non autenticato tenta di firmare, viene condotto alla schermata di login (Figura 3). Per tornare alla lista di preview, l'utente può cliccare su "Torna Indietro", in alto a sinistra. Se l'iniziativa è stata importata da una piattaforma esterna, il pulsante per firmare conduce l'utente alla piattaforma di provenienza, ed ha il seguente aspetto:



4.8 Dashboard personale

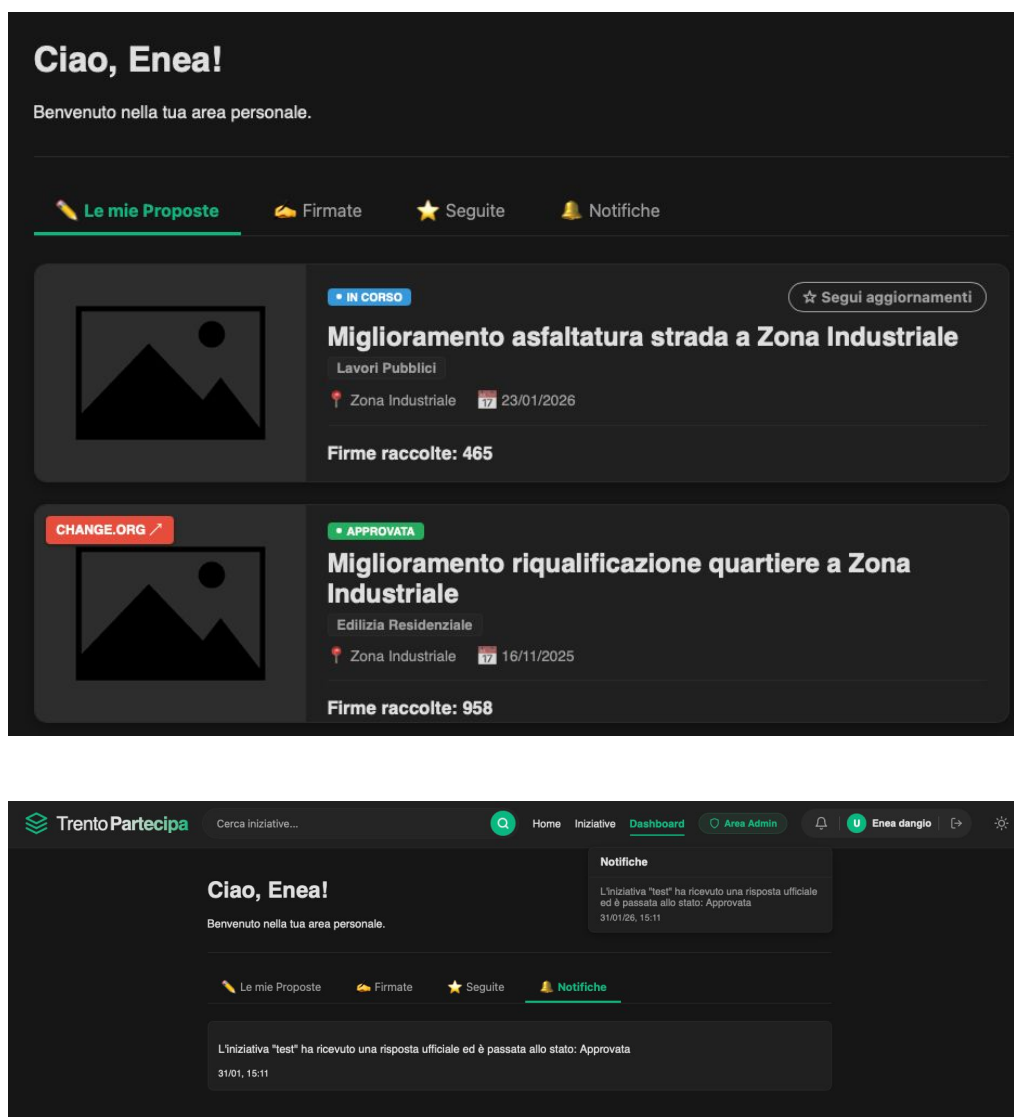


Figura 8: Dashboard personale

In Figura 8 sono raffigurate le schermate della dashboard personale di un cittadino, dalla quale può visualizzare le iniziative da lui create, firmate e seguite, oltre alle notifiche che gli sono giunte (visualizzabili anche cliccando sull'icona della campanella nell'header).

4.9 Area Admin

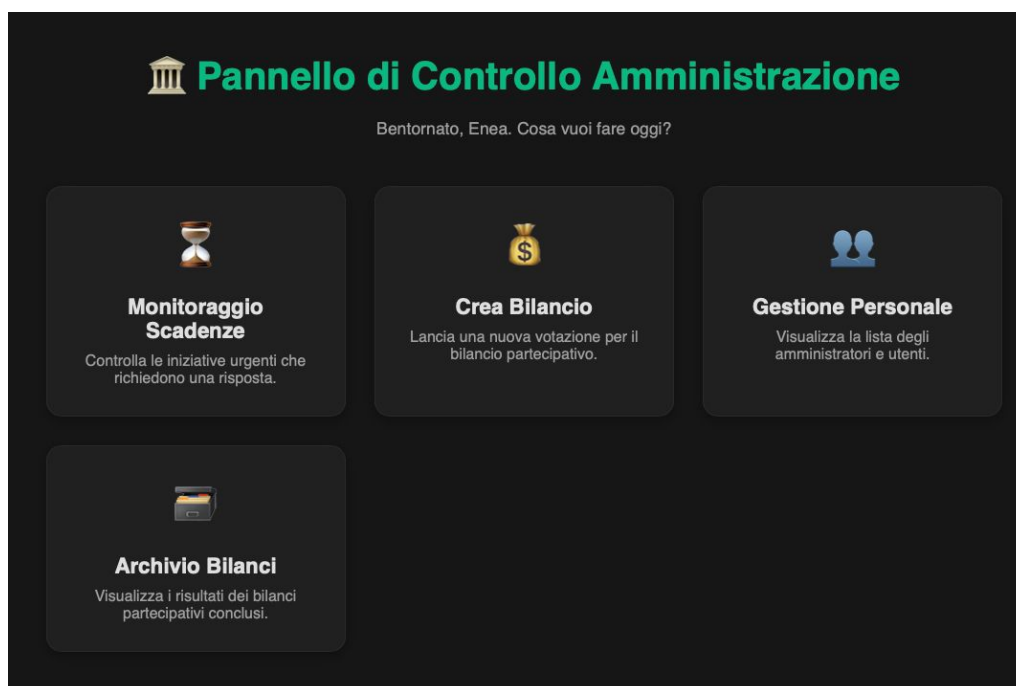


Figura 9: Area Admin

In Figura 9 è raffigurata l'area admin, accessibile dagli utenti amministratori. L'area admin consiste in un pannello di controllo dal quale l'utente può scegliere che attività svolgere. Selezionando una tra le quattro aree l'utente sarà ricondotto alla pagina dedicata a quell'attività. Le pagine in questione sono descritte di seguito (Figure 10, 13, 14, 15).

4.10 Schermata di monitoraggio delle iniziative

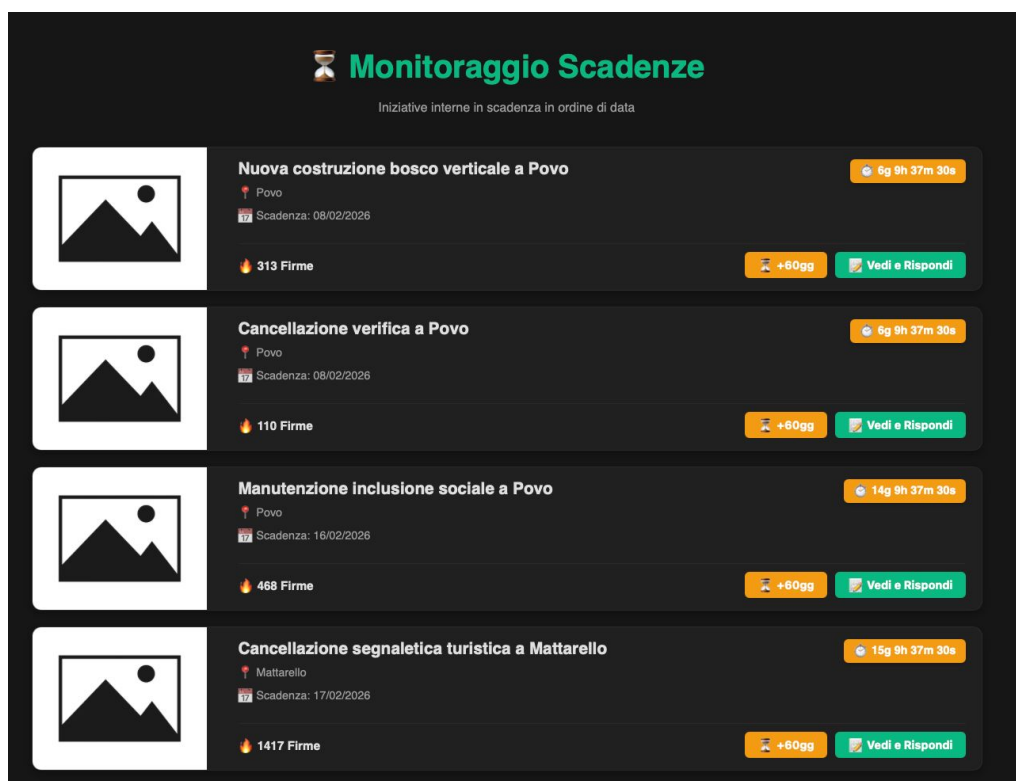


Figura 10: Schermata di monitoraggio delle iniziative

In Figure 10 è raffigurata la schermata da cui l'utente amministratore può monitorare le iniziative in corso, in attesa di risposta. Ogni iniziativa riporta un timer che indica il tempo rimasto prima che scada. Sono inoltre presenti, per ogni iniziativa, un pulsante per prorogarne la data di scadenza di 60 giorni e un pulsante per visualizzarla e scrivere una risposta (Figura 11).

4.11 Schermata di compilazione di una risposta del Comune

The screenshot shows a web interface for submitting an official response from a municipality. At the top, a dark grey box titled 'Descrizione Originale' contains the text: 'Proposta per cancellazione di verifica situato in zona Povo. Questa iniziativa mira a migliorare la qualità della vita per la categoria Test Category.' Below this, a large light blue box with a rounded border is titled 'Risposta Ufficiale dell'Amministrazione' with a building icon. Inside this box, the text 'Compila questo modulo per concludere l'iniziativa.' is followed by the 'Esito Finale:' section, which has two radio button options: 'Accogli Proposta' (selected) and 'Respingi Proposta'. Below this is the 'Motivazione Ufficiale (Obbligatoria):' section with a text area containing the placeholder 'Scrivi qui la risposta dettagliata del Comune...'. The 'Allegati (Documenti ufficiali, Delibere):' section features a file upload button labeled 'Clicca qui per aggiungere documenti' and a status message 'Nessun allegato selezionato.' at the bottom. Two large buttons are at the very bottom: an orange 'Estendi Scadenza (+60gg)' button and a blue 'Invia Risposta Definitiva' button.

Figura 11: Schermata di compilazione di una risposta del Comune

In Figura 11 è raffigurata la schermata da cui l'utente amministratore può compilare una risposta per un'iniziativa. Il box con i campi da compilare appare all'amministratore sotto le informazioni presenti nella pagina di dettaglio dell'iniziativa selezionata. L'amministratore deve selezionare l'esito finale dell'iniziativa ("Approvata" o "Respinta"), scrivere una motivazione ufficiale e allegare eventualmente dei documenti. Tramite un apposito pulsante può poi pubblicarla. Anche qui è presente un pulsante per prorogare la data di scadenza dell'iniziativa.

Una volta pubblicata, la risposta apparirà agli altri utenti nella pagina di dettaglio dell'iniziativa (Figura 7), come nella figura sotto:

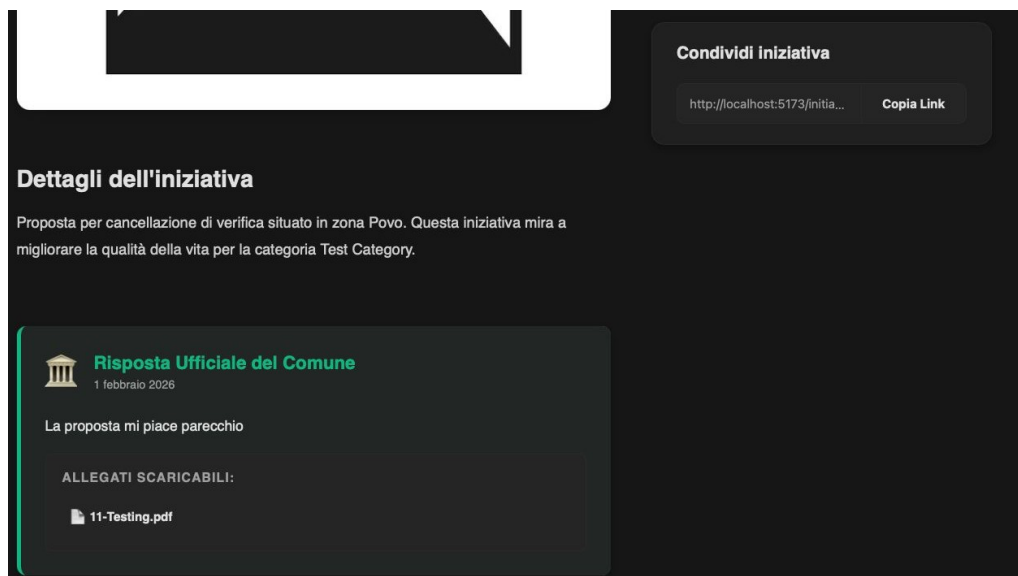
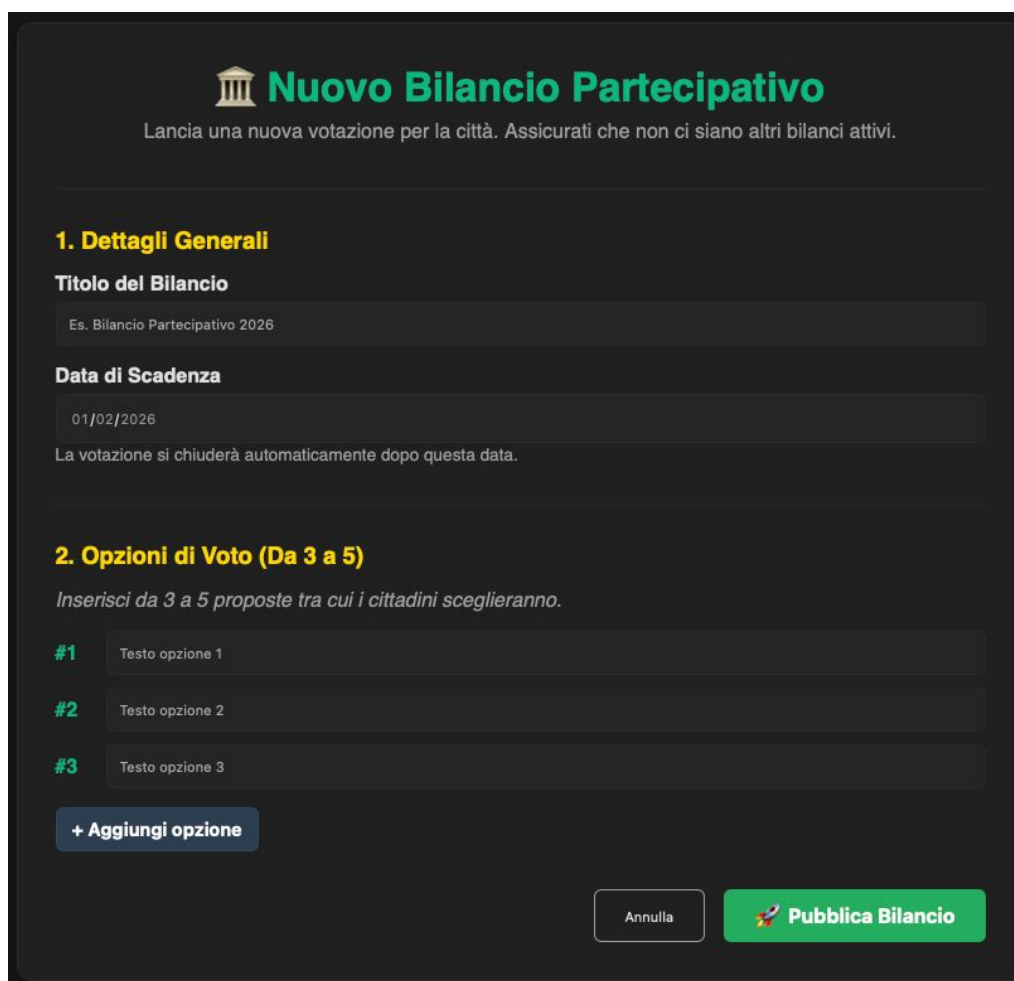



Figura 12: Risposta del Comune all'iniziativa

4.12 Schermata di creazione del bilancio partecipativo



 **Nuovo Bilancio Partecipativo**

Lancia una nuova votazione per la città. Assicurati che non ci siano altri bilanci attivi.

1. Dettagli Generali

Titolo del Bilancio

Es. Bilancio Partecipativo 2026

Data di Scadenza

01/02/2026

La votazione si chiuderà automaticamente dopo questa data.

2. Opzioni di Voto (Da 3 a 5)

Inserisci da 3 a 5 proposte tra cui i cittadini sceglieranno.

#1 Testo opzione 1

#2 Testo opzione 2

#3 Testo opzione 3

+ Aggiungi opzione


Annulla  **Pubblica Bilancio**

Figura 13: Schermata di creazione del bilancio partecipativo

In Figura 13 è raffigurata la schermata da cui l'utente amministratore può compilare i campi relativi a un nuovo bilancio partecipativo che intende pubblicare: questi campi, digitabili nei form, includono il titolo, la data di scadenza e un certo numero di opzioni compreso tra 3 e 5. Un apposito pulsante permette di aggiungere un'opzione, se possibile. Sono poi presenti dei pulsanti per annullare o confermare l'operazione.

4.13 Schermata di gestione del personale

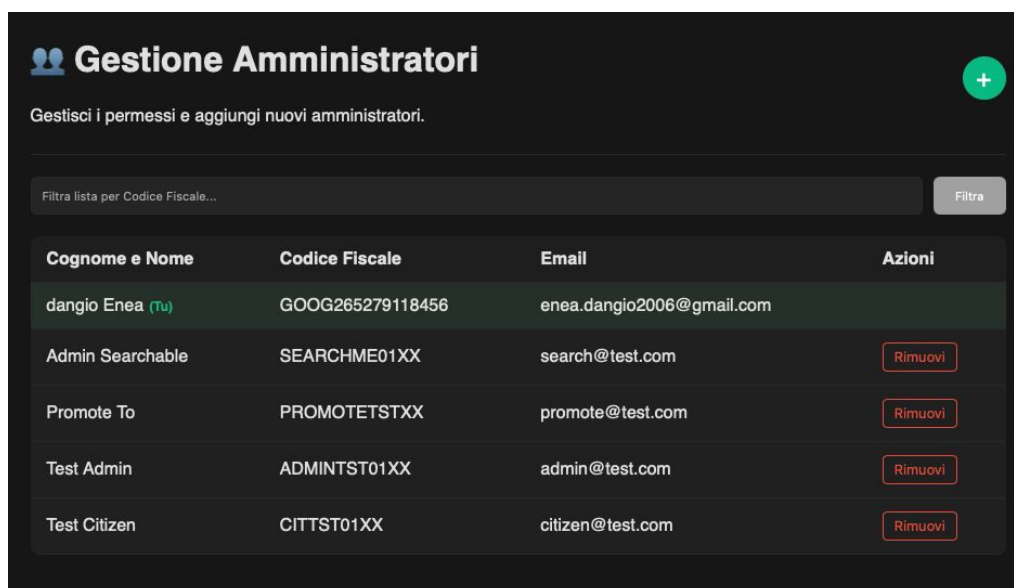


Figura 14: Schermata di gestione del personale

In Figura 14 è raffigurata la schermata dalla quale l'utente amministratore può visualizzare la lista completa degli utenti amministratori con le loro informazioni (cognome, nome, codice fiscale, email). Una barra di ricerca permette di digitare un codice fiscale per verificarne la presenza nell'elenco. Per ogni utente nella lista, ad esclusione dell'utente stesso, è presente un pulsante "Rimuovi" che serve per revocare i privilegi da amministratore ad un certo utente. Per aggiungere un nuovo amministratore (promuovendo un cittadino o creando una pre-autorizzazione), l'utente può cliccare il pulsante raffigurante il "+", in alto a destra: così facendo apparirà un box (come nella figura sotto) dove l'utente potrà digitare il codice fiscale dell'utente che intende promuovere. Se l'utente ricercato non è presente nel database appare un box per confermarne la pre-autorizzazione.



4.14 Archivio dei bilanci partecipativi



Figura 15: Archivio dei bilanci partecipativi

In Figura 15 è raffigurata la schermata da cui l'utente amministratore può consultare l'archivio dei bilanci partecipativi conclusi. Ogni bilancio partecipativo riporta titolo, data di scadenza, numero di opzioni e un pulsante per visualizzarne i risultati. Cliccandolo si apre una schermata come la seguente:



Figura 16: Risultati di un bilancio partecipativo concluso

In Figura 16 sono raffigurati i risultati di un bilancio partecipativo concluso: vengono riportati il titolo del sondaggio, la data di scadenza, il numero totale dei voti e le opzioni. A fianco ad ogni opzione sono riportati il numero di voti e la percentuale sul totale, rispecchiata dalla barra verde sotto l'opzione. Per tornare all'archivio, l'utente può cliccare sul pulsante in basso a destra.

5 Deployment

Un'istanza del backend è ospitata sulla piattaforma `render.com` ed è disponibile al link <https://trentopartecipa.onrender.com>. Il database relazionale invece è attualmente ospitato sulla piattaforma DigitalOcean che grazie all'account education di github abbiamo potuto utilizzare gratuitamente. Il frontend invece è deployato separatamente al link trentopartecipa.me.

L'utilizzo del piano gratuito (Free Tier) su Render comporta una limitazione prestazionale intrinseca nota come **Cold Start**: in caso di inattività prolungata (superiore ai 15 minuti), la piattaforma sospende l'istanza per risparmiare risorse. Di conseguenza, la prima richiesta successiva a un periodo di pausa potrebbe richiedere circa 50-60 secondi per il riavvio del servizio; le richieste successive avverranno invece con latenza standard.

Per superare invece le restrizioni funzionali (rete e storage) imposte dalla piattaforma, sono state adottate le seguenti soluzioni architetturali:

- **Migrazione da SMTP a API HTTP (Resend):** Poiché Render blocca le porte SMTP in uscita, abbiamo sostituito l'invio tradizionale delle email con la libreria **Resend**. Questo servizio permette di inviare i codici OTP per il login tramite chiamate API HTTP (REST), aggirando le restrizioni di rete.
- **Verifica del Dominio e Configurazione DNS:** Per superare il blocco di sicurezza di Resend (che impedisce l'invio a indirizzi terzi senza verifica), abbiamo registrato il dominio trentopartecipa.me tramite *Namecheap*. Abbiamo quindi configurato i record DNS **SPF** e **DKIM** per autorizzare l'invio delle email e i record **A** per puntare agli indirizzi IP di GitHub Pages, servendo così anche il frontend dal dominio personalizzato.
- **Persistenza dei Media (Cloudinary):** Dato che il filesystem di Render è effimero (i file locali vengono persi allo spegnimento dell'istanza), le immagini caricate dagli utenti vengono salvate sul cloud di **Cloudinary**, memorizzando nel database solo l'URL remoto sicuro.

Configurazione CI/CD La configurazione CI/CD basata sulle GitHub Actions esegue il deploy del branch main solo quando vengono superati tutti i test con successo.

5.1 Accesso docenti

I docenti potranno testare interamente l'applicazione sia tramite il normale login con google (che permette l'accesso come utente cittadino) sia tramite un accesso hardcoded utilizzando il codice secret. Sarà poi possibile scegliere il tipo di utente nell'applicativo al link: <https://trentopartecipa.me/teacher-login>.

È importante specificare che questo tipo di accesso è a solo scopo di testing, in modo da poter velocizzare la revisione di tutte le funzionalità per utenti che non hanno direttamente accesso al database.

Secret: x7K9pLm2Q5nRt8vW4yB1jD6fG3hC0zXs