

Polytechnic University of Bucharest  
Faculty of Automatic Control and  
Computer Science

## **EventReport**

Dezvoltarea Aplicațiilor pentru Internet

Radu Catrangiu  
SSA

<b>Sinopsis</b>	<b>3</b>
<b>Introducere</b>	<b>4</b>
Problema	4
Obiectivul	4
<b>Alte solutii existente</b>	<b>5</b>
DSU	5
Seism Alert	5
INOVAGRIA Meteo	5
<b>Arhitectura aplicatiei</b>	<b>6</b>
<b>Detalii de implementare</b>	<b>7</b>
Backend	7
Frontend	8

## Sinopsis

EventReport este un proiect care ajuta comunitatea in a raporta evenimente majore sau dezastre naturale cum ar fi cutremure, incendii, inundatii, accidente majore. Este o aplicatie web in care oricine poate semnala un astfel de eveniment. Aplicatia este creata pornind de la orasul Bucuresti dar poate fi cu usurinta extinsa. Punctele forte ale aplicatiei vin din usurinta de folosire si interfata intuitiva.

# Introducere

În prezent, în România nu există niciun sistem public central de monitorizare al dezastrelor care să fie accesibil și de cetățeni. În momentul de față, pentru a afla despre eventualele dezastre sau evenimente majore, cetățenii au ca posibile surse de informații:

- televiziunea, radioul
- rețelele de socializare
- site-uri de știri
- alerte mobile de la autorități

Toate aceste surse prezintă numeroase dezavantaje: nu este garantat că dacă un dezastru are loc, informația poate fi aflată rapid prin intermediul televiziunii / radioului, pentru a afla informații de pe site-urile de știri trebuie investit timp pentru a filtra eventualele informații, iar alertele mobile trimise de autorități avizează doar persoanele aflate în proximitatea evenimentului.

## Problema

Problema pe care proiectul dorește să o rezolve constă în lipsa unei platforme de raportare de incidente de către cetățeni rapid și ușor. În momentul de față, nu am identificat existența niciunei aplicații sau al unui loc de centralizare pentru evenimente de acest tip, care să fie accesibil și ușor de folosit dedicat cetățenilor. Aceștia nu au loc special în care ar putea raporta direct atât autorităților, cât și altor cetățeni declansarea unui eveniment cu un impact major asupra orașului.

## Obiectivul

Obiectivul acestui proiect este de a dezvolta o platformă cu ajutorul căreia cetățenii să poată raporta și observa evenimente. Aplicația are două funcționalități importante: un utilizator poate raporta un eveniment major sau poate observa un eveniment (raportat de către alt utilizator). Această aplicație poate fi utilizată și de autorități pentru a agrega și monitoriza situațiile de urgență din oraș.

Ne propunem să cream o platformă accesibilă și de încredere. Punctul forte al acestei aplicații vine din simplitatea și din complexitatea scăzută din punct de vedere al utilizatorului, astfel ca persoane de toate vârstele și fără vreo pregătire în avans o pot utiliza.

## Alte solutii existente

### DSU

Această aplicație este oferită de Departamentului pentru Situații de Urgență cu scopul de a pune la dispoziția cetățenilor un instrument rapid de informare și de raportare a situațiilor critice și a dezastrelor la care aceștia sunt martori. Dezavantajele aplicației vin din faptul că cel care gestionează toate evenimentele raportate este administratorul aplicației, iar cetățenii nu au acces la evenimente raportate de alți utilizatori, astfel că se introduce un timp suplimentar până când cetățenii pot afla informații despre incidente.

O altă lipsă a aplicației constă în faptul că nu se poate vedea dacă există un incident într-o anumită locație. De asemenea, aceasta aplicație nu este disponibilă și din browser limitând accesul cetățenilor care nu au telefonul mobil accesibil în momentul în care au nevoie să folosească aplicația.

Astfel, putem concluziona că această aplicație, deși este foarte utilă cetățenilor, nu rezolvă problema enunțată.

### Seism Alert

Această aplicație semnalează utilizatorilor când are loc un cutremur în zona Vrancea de peste 4.5 grade pe scara Richter. Aplicația nu rezolvă problema enunțată deoarece nu acoperă decât o categorie restrânsă a dezastrelor naturale (cutremurele) iar clienții nu au posibilitatea de a raporta niciun eveniment. De asemenea, aplicația funcționează doar dacă utilizatorii țin pagina web deschisă continuu, ceea ce o face foarte inaccesibilă și incomodă.

### INOVAGRIA Meteo

Aplicația INOVAGRIA oferă informații și alerte meteo și agrometeo detaliate de la Administrația Națională de Meteorologie. La fel ca în cazul aplicației destinate seismelor, INOVAGRIA nu oferă posibilitatea raportării de incidente de către utilizatori și acoperă o gamă restrânsă de incidente (evenimente meteo). Așadar aplicația nu este o soluție pentru problema enunțată.

## Arhitectura aplicatiei

In sectiunea curenta se va prezenta arhitectura aplicatiei propuse.

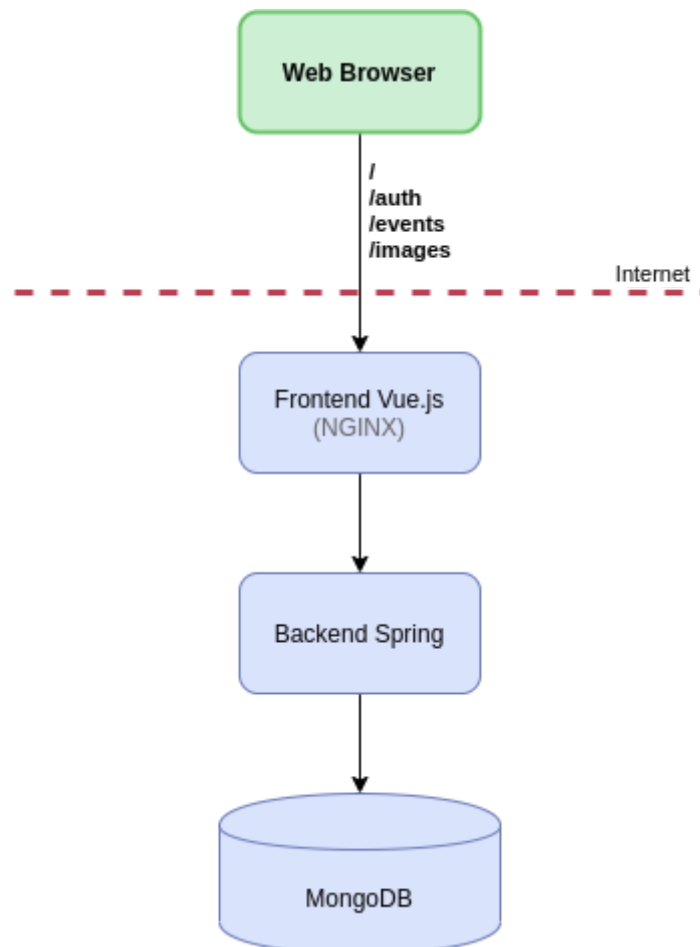


Fig. 1: Diagrama de arhitectura a aplicatiei

Dupa cum se poate observa in diagrama, platforma dezvoltata are doua componente principale:

- server care va stoca in baza de date evenimentele raportate
- o interfata web care permite vizualizarea evenimentelor raportate pe o harta, dar care permite si raportarea de evenimente noi

Serverul care se ocupa de evenimentele raportate este format la randul sau din mai multe componente, acestea fiind toate impachetate in containere de Docker:

- containerul de **frontend** este un server de NGINX care pentru calea '/' serveste fisierele statice de frontend (interfata web propriu-zisa); pentru căile '/auth', '/event' și '/image/\*' are rol de proxy catre container-ul 'backend'

- **serviciul de backend** se ocupa de toate procesarile si interactiunea legata de evenimente și se ocupa de toate interactiunile cu imagini din cadrul aplicatiei (postarea si servirea lor);
- **baza de date** care stocheaza toate informatiile legate de evenimentele raportate

Un eveniment este caracterizat de:

- timp de raportare
- locatie
- descriere
- poza justificativa
- eticheta de clasificare

## Detalii de implementare

### Backend

Backend-ul aplicatiei este scris in **Java Spring** si **MongoDB** ca baza de date. Este compus din 2 servicii, fiecare in propriul container de Docker: serviciul de backend care se ocupa de management-ul evenimentelor, autentificare și al imaginilor, iar serviciul de frontend care este efectiv un container de NGINX ce se ocupa de servirea resurselor statice (fisierelor de frontend: fisiere de tipul html, css, javascript) si de rutarea cererilor cu calea '/events', '/auth' și '/image' catre backend.

#### Endpointurile expuse de backend-ul aplicatiei:

- **GET /auth/login** primeste email si parola
  - creaza o sesiune noua, intoarce id-ul utilizatorului, statusul de admin si tokenul de sesiune
- **GET /auth/token**
  - primeste tokenul de sesiune si intoarce id-ul utilizatorului, statusul de admin si email-ul utilizatorului
- **POST /auth/create**
  - primeste email si parola si creaza un user nou ce poate fi transformat in 'admin' ulterior
  - intoarce 200Ok
  - intoarce 403Forbidden daca utilizatorul introduce date invalide sau daca exista deja un utilizator cu acest email
- **GET /events**
  - intoarce lista de evenimente sortata cronologic
- **POST /events**
  - primeste titlu, descriere, tag, location si image\_id
  - poate primi eventual ca parametru un token de autentificare pentru a atribui evenimentul nou unui utilizator
  - la postare trimite cate un email cu datele evenimentului catre toti administratorii daca id-ul imaginii corespunde cu o imagine din baza de date.

- pentru fiecare administrator este pornit un nou thread ce se ocupa de compunerea si trimiterea mail-ului, astfel încât răspunsul cererii să nu fie întârziată
  - actualizeaza imaginea din baza de date adaugand event\_id-ul si eliminand campul ce specifica data de expirare, astfel evitând ștergerea automata a imaginii.
  - intoarce 200Ok si evenimentul adaugat in baza de date alaturi de un id (claim\_id) care poate fi utilizat pentru asocierea dintre un eveniment si un utilizator
    - daca request-ul este facut cu token de autentificare ca parametru, claim\_id va fi null in raspuns
    - id-ul claim\_id este intors o singura data, la crearea evenimentului
- **PUT /events/:id**
  - primeste un token de autentificare si un boolean resolved
    - token-ul de autentificare trebuie sa apartina unui administrator
    - daca documentul cu id-ul primit exista si token-ul este valid, documentul este actualizat folosind boolean-ul primit.
- **DELETE /events/:id**
  - primeste un token de autentificare
    - token-ul de autentificare trebuie sa apartina unui administrator sau celui care a postat evenimentul
    - daca documentul cu id-ul primit exista si token-ul este valid, documentul este sters din baza de date.
- **POST /events/claim**
  - primeste un token de autentificare si o lista de perechi (event\_id, claim\_id)
  - seteaza proprietatea owner\_id pe fiecare eveniment pentru care claim\_id-ul din request corespunde cu cel din obiectul de tip eveniment
    - de asemenea, este setat claim\_id pe null pentru a face imposibila schimbarea owner-ului unui eveniment

#### Endpoint-urile expuse de image-service:

- **POST /image/upload**
  - incarca o imagine si o stocheaza in baza de date
  - intoarce id-ul imaginii
- **GET /image/:id**
  - obtine din baza de date imaginea cu id-ul primit, o trimite ca raspuns in format binar, astfel incat aceasta poate fi interpretata ca o imagine.

## Frontend

Interfata web este simpla si intuitiva. Aceasta se deschide cu harta orasului Bucuresti, dar utilizatorul poate cu usurinta muta focusul hartii pe alta zona. Poate filtra evenimentele cautând după anumite cuvinte cheie, lucru ce va actualiza lista si harta in functie de rezultate.



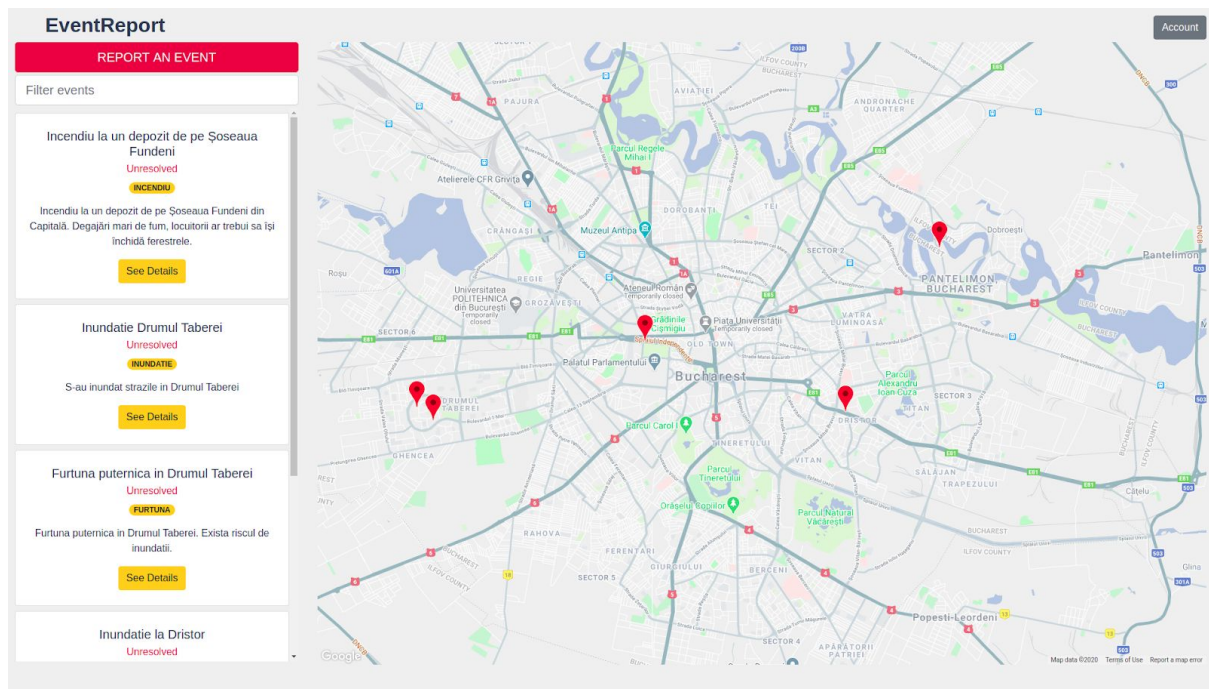


Fig.2: Interfata aplicatiei web

Utilizatorii se pot inregistra sau autentifica in aplicatie folosind butonul **Account**. Utilizatorii normali pot adauga evenimente pe care le pot sterge ulterior. Utilizatorii cu drepturi de administrare sunt cei care pot marca un eveniment ca fiind rezolvat sau il pot sterge. O alta caracteristica speciala este faptul ca de fiecare data cand un eveniment este raportat, administratorii sunt notificati prin intermediul adresei de email.

Fig. 2: Fereastra de inregistrare/autentificare



Fig. 3: Sumarizarea unui eveniment vazut de un admin

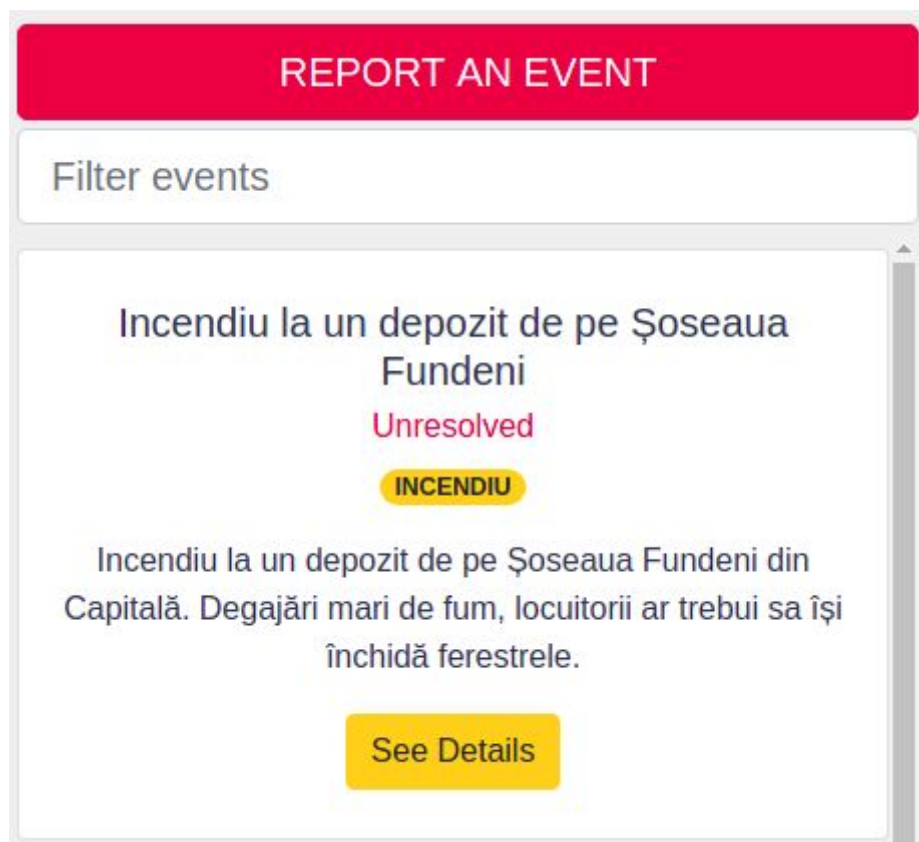


Fig. 4: Sumarizarea unui eveniment vazut de un utilizator normal

Pentru a raporta un eveniment, utilizatorul trebuie sa completeze un formular in care va mentiona un titlu sugestiv care sa sumarizeze evenimentul raportat, tipul de eveniment, o descriere a acestuia in care poate sa mentioneze detalii suplimentare si o poza legata de evenimentul raportat. Locatia poate fi detectata automat sau utilizatorul poate selecta

manual locatia alegand un loc de pe harta. Daca utilizatorul nu este autentificat, aplicatia stocheaza in browser detaliile pentru revendicarea evenimentelor la login (perechile claim\_id - event\_id).

Event Title

Event Title

Event Type

Choose...

Event Description

Event Picture

Choose file

Browse

Pick Location of Event

Get Browser Location

Pick Location on Map

REPORT EVENT

Fig. 5: Formularul pentru completarea unui eveniment

Incendiu la un depozit de pe Șoseaua Fundeni

×

Description:

Incendiu la un depozit de pe Șoseaua Fundeni din Capitală.  
Degajări mari de fum, locuitorii ar trebui sa își închidă ferestrele.

Report Date:

17.05.2020, 17:55:13

Image:



Close

Fig. 6: Prezentarea unui eveniment

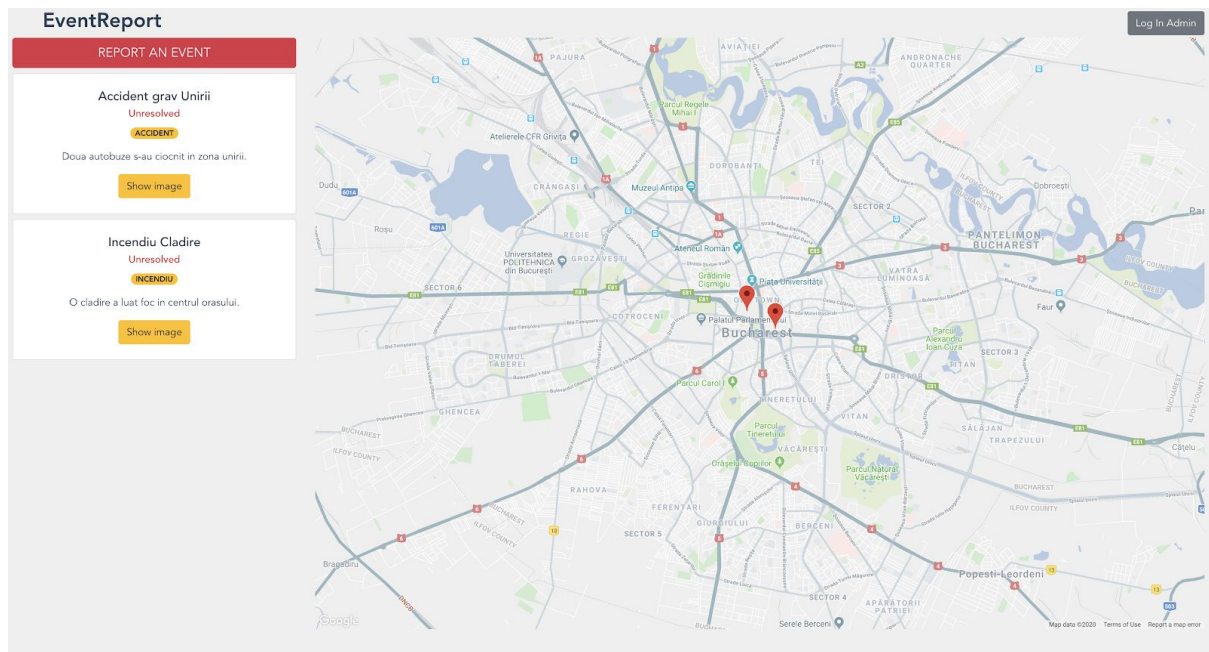


Fig. 7: Interfata aplicatiei web cu evenimente raportate

Odata raportate, evenimentele sunt afisate intr-o lista in partea stanga a ecranului si, de asemenea, sunt marcate pe harta printr-un pin rosu. Cand un pin este selectat, evenimentul corespunzator este marcat vizual.

Interfata a fost dezvoltata in **Vue.js**, requesturile catre backend realizandu-se folosind **Axios** (o biblioteca de Javascript cu ajutorul careia se pot face request-uri HTTP).

Comunicatia intre componente, precum lista de evenimente si harta, are loc printr-un “event bus” (o alta instanta de Vue.js care a ca singur scop comunicatia folosind componenta interna de evenimente - se emit si se pot asculta dupa evenimente).

La deschiderea paginii:

- harta utilizata este de la Google Maps cu niste aspecte personalizate (culori).
- butoanele de administrare sunt afisate doar daca este indeplinita o conditie, lucru implementat folosind directiva “v-if” din Vue.js.
- componenta EventList verifica la incarcare daca utilizatorul este administrator si trimite catre fiecare subcomponenta un boolean “adminUser”.
  - EventList este o componenta care are ca sub-componente module de tipul EventCard
  - EventCard-urile sunt elemente implementate pentru a afisa în mod stilizat lista de evenimente din stanga ecranului
- dupa incarcarea hărții are loc un GET către calea ‘/events’ pentru a se obtine toate evenimentele.
- fiecare EventCard, la incarcare emite un eveniment catre hartă pentru adaugarea pin-ului impreuna cu listener-ul de pe punct;

- atunci cand se da click pe punct este emis un eveniment catre EventList pentru a face scroll și highlight pe EventCard-ul corespunzator punctului pe care a fost facut click. Animatia de scroll este facuta folosind **jQuery**, iar highlight-ul este implementat prin adaugarea clasei "highlighted" pe card-ul corespunzator, timp de o secunda.

Aplicatia are doua modale:

- login pentru utilizatori
  - efectueaza un apel GET catre '/auth/login'
  - daca raspunsul este 200Ok, este stocat in cookie-uri "login\_token"-ul, utilizat pentru a realiza apel-urile ce necesita drepturi de administrator
  - se memoreaza detaliile utilizatorului
  - daca exista evenimente raportate anterior, este facut apelul '/events/claim' pentru a le revendica pentru utilizator.
- poza si detaliile unui eveniment
  - afiseaza imaginea prin setarea proprietatii "src" pe tag-ul <img> cu un link generat in mod individual (folosind un image\_id)

Componenta EventList poate comuta intre lista efectiva de EventCards si un NewEventCard la apasarea butonului de 'Report an event' / 'Back to list'.

- Componenta NewEventCard permite introducerea datelor despre eveniment, alegerea unui punct pe harta prin emiterea unui event, către componenta GoogleMap, ce permite utilizatorului sa dea un singur click pentru a inregistra o noua pozitie, sau prin emiterea unui alt event ce transmite coordonatele obtinute din browser.
- la incarcarea unei imagini are loc automat un POST catre '/image/upload' care incarca imaginea si intoarce un image\_id care expira in 5 minute. Dupa trecerea celor 5 minute imaginea este stearsa din baza de date daca nu este asociata unui eveniment.

## Concluzii

In urma acestui proiect, a fost realizata o platforma de raportare de evenimente simpla, intuitiva si usor de folosit. Aceasta rezolva problema enuntata, oferind posibilitatea centralizarii evenimentelor si dezastrelor majore intr-un singur loc, accesibil cetatenilor. Astfel, daca cineva este interesat de o anumita zona, poate cu usurinta verifica daca s-a produs un eveniment sau poate rapid sa raporteze un eveniment.