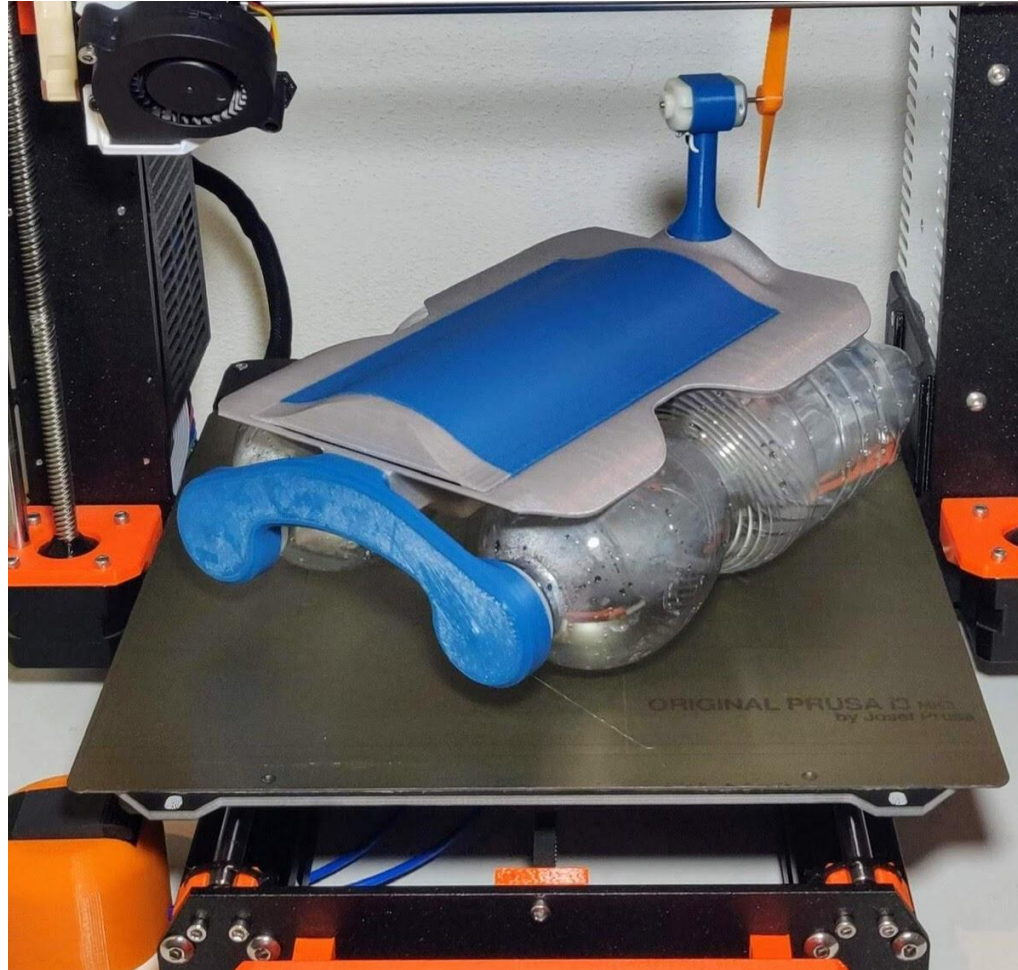


# Database barchette

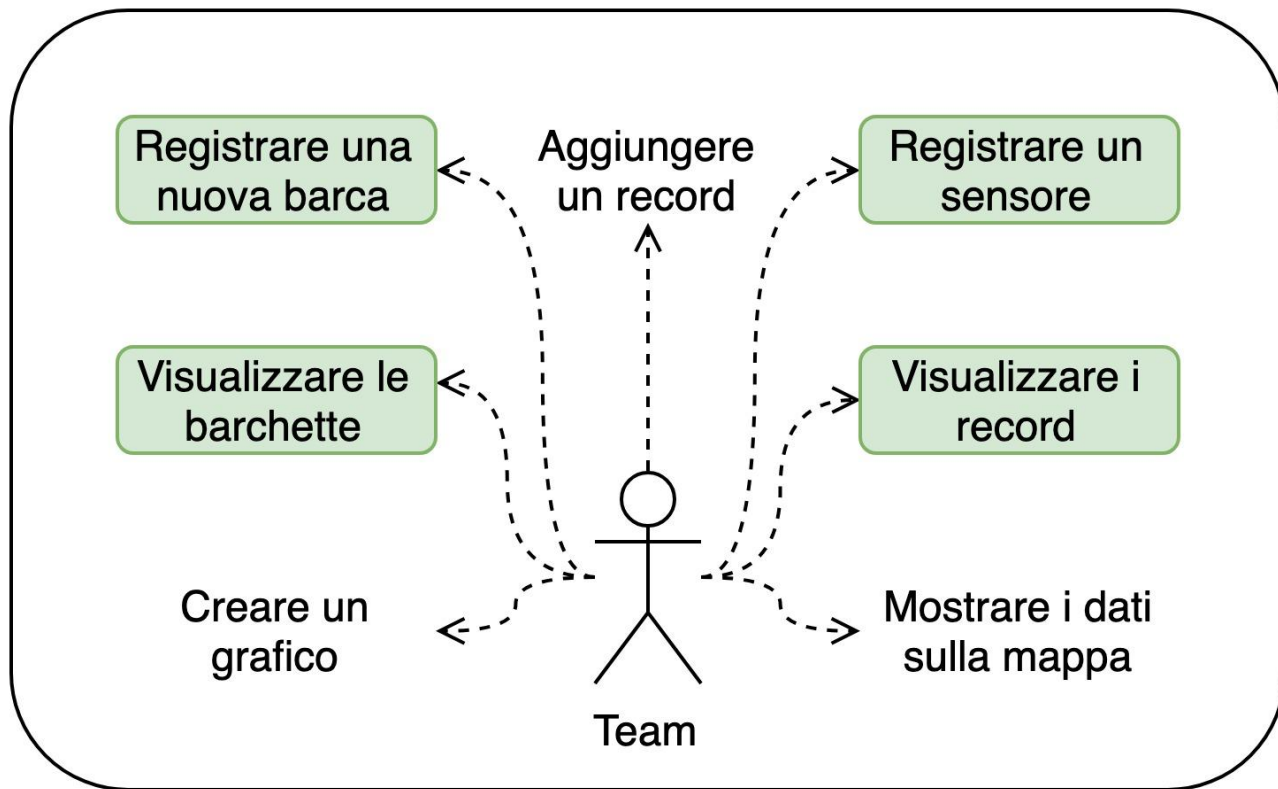
Alberto Nidasio - Stefano Vassallo

# Problema

Il team Specchi d'Acqua, che parteciperà alle prossime olimpiadi di robotica a Genova, ha bisogno di salvare i dati ambientali raccolti dalle proprie barchette (temperatura, umidità, ecc...).



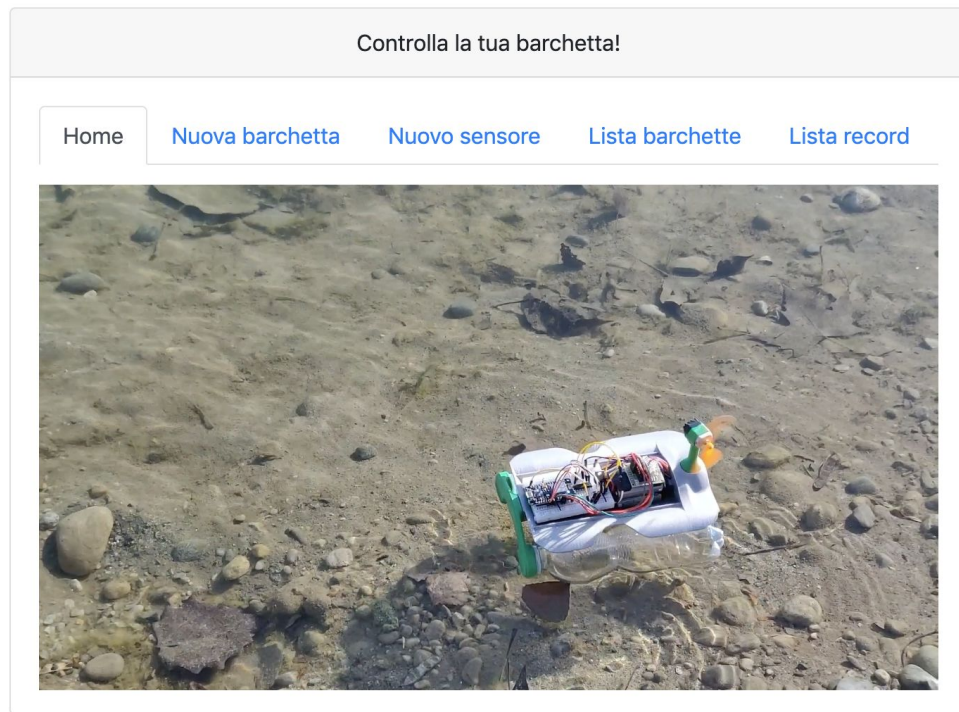
# Casi d'uso



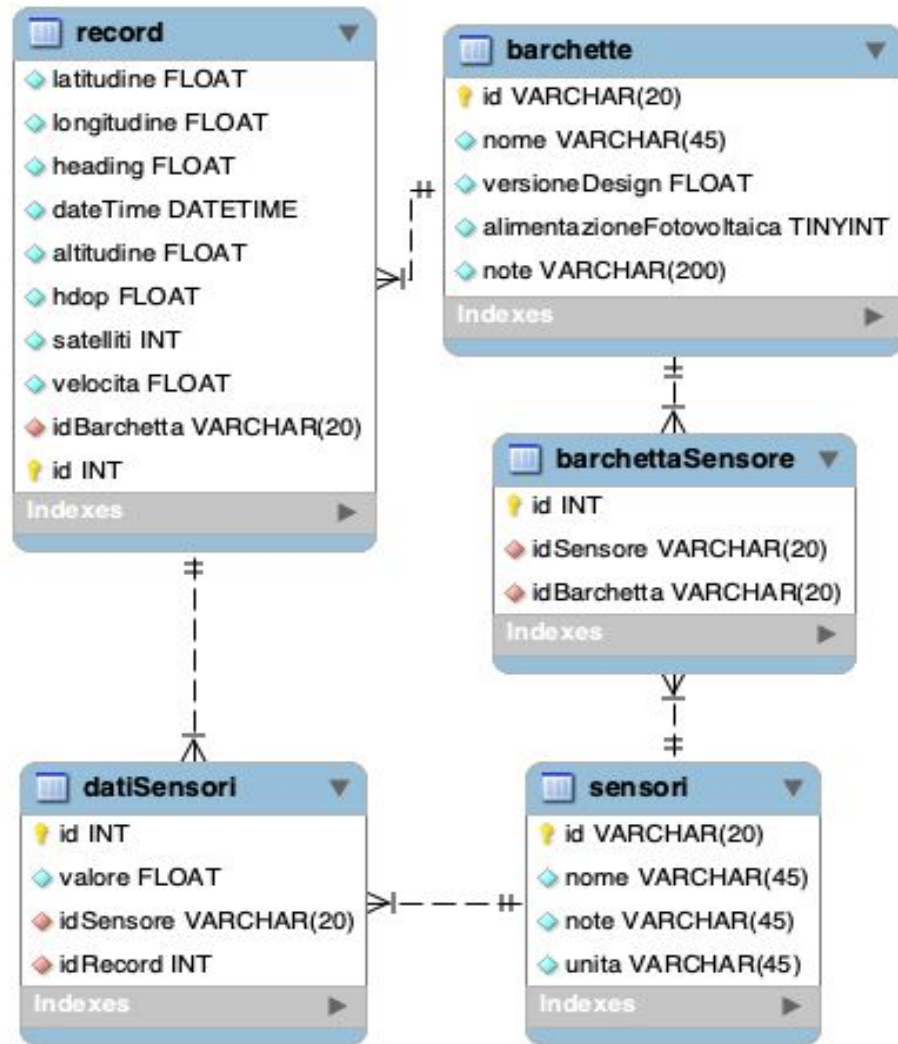
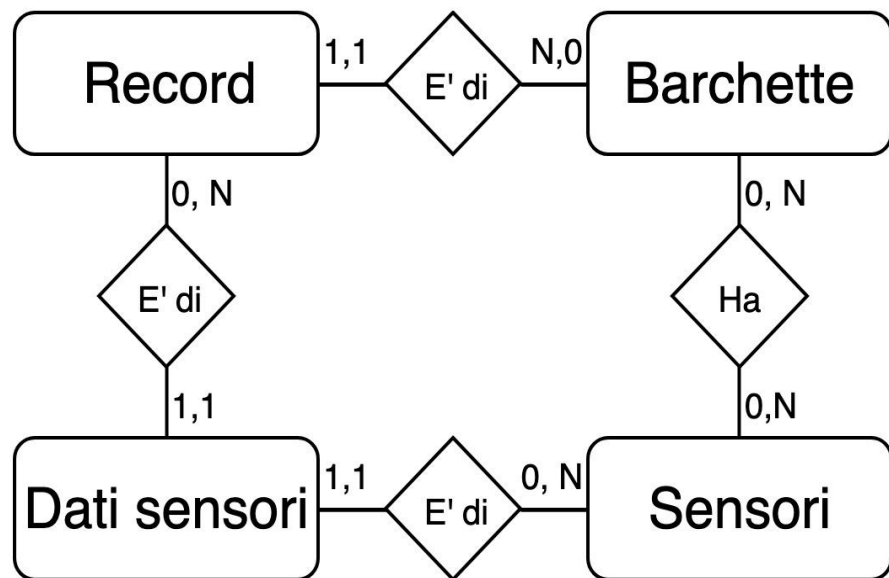
# Che cosa abbiamo fatto

Il progetto si divide in 3 parti:

- Database;
- Server REST;
- Sito web;



# Schema database



# Server

```
17 //preparo express
18 const app = express()
19 app.use(express.json())
20 app.use(cors({origin: '*'}))
21
22 //rendo disponibile la pagina web nella root dell'url
23 app.use('/', express.static('website'))
24
25 //SEZIONE RECORD
26
27 //lista record con i dati dei sensori
28 app.get('/api/record', async (req, res) => {
29   console.log(req.method, req.url, req.query, req.body)
30
31   try {
32     return res.send(await database.ottieniRecord(req.query))
33   } catch (errore) {
34     return res.status(500).send(errore.message)
35   }
36 })
```





# Client - GET

Controlla la tua barchetta!

[Home](#)   [Nuova barchetta](#)   [Nuovo sensore](#)   [Lista barchette](#)

[Lista record](#)

#	Nome	V. Design	Fotovoltaico
1	barca1	1	0
2	barca2	2	0
3	barca3	3	0
4	barca4	4	0
5	test	1	1

```
//funzione per caricare le barchette nella tabella
function caricaBarchette() {
    const http = new XMLHttpRequest()
    const url = 'http://localhost/api/barchette'
    http.open('GET', url)
    http.onload = (e) => {
        var data = JSON.parse(http.responseText)
        $('#tabella-barchette').bootstrapTable({
            columns: [{
                field: 'id',
                title: '#'
            }, {
                field: 'nome',
                title: 'Nome'
            }, {
                field: 'versioneDesign',
                title: 'V. Design'
            }, {
                field: 'alimentazioneFotovoltaica',
                title: 'Fotovoltaico'
            }
        ],
            data
        })
    }
    http.send()
}
```

# Client - POST

```
//funzione per registrare un nuovo sensore
function aggiungiSensore() {
    //controllo i dati
    if(document.getElementById("nome").value == "" ||
        document.getElementById("nota").value == "") {
        //mostro un avviso all'utente
        alert("Compilare tutti i campi!");
    } else{
        const http = new XMLHttpRequest()
        const url = 'http://10.0.98.125/api/sensori'
        http.open('POST', url,true)
        http.setRequestHeader("Content-Type", "application/json");
        var sensore = {
            nome: document.getElementById("nome").value,
            note: document.getElementById("nota").value
        }
        http.send(JSON.stringify(sensore));
    }
}
```

Controlla la tua barchetta!

[Home](#) [Nuova barchetta](#) [Nuovo sensore](#) [Lista barchette](#) [Lista record](#)

Inserire il nome del sensore:

Inserisci una nota

Aggiungi



# Struttura della presentazione

1. Descrizione problema (slide 1 e 2) - Nidasio
2. Definizione delle funzionalità da realizzare / casi d'uso (slide 3 e 4) - Vassallo
3. Analisi dei dati e schema DB (slide 5) - Nidasio
4. Implementazione server (slide successive o codice) - Nidasio
5. Implementazione client (slide successive o codice) - Vassallo