



PROGETTO GESTIONE PIZZERIA

ALFIERI, PAGGI, ZILIOLI





Indice Unità

- 1 Analisi requisiti
- 2 Overview del problema
- 3 Mockup
- 4 Progettazione concettuale
- 5 Progettazione logica
- 6 Progettazione fisica



ANALISI REQUISITI

REQUISITI FUNZIONALI

- Il sistema consente al responsabile di creare e inserire nuovi articoli all'interno del menù.
- Il gestore può consultare in qualsiasi momento la lista completa dei prodotti disponibili nel menù.
- È possibile per il gestore rimuovere definitivamente un prodotto dal menù.
- Il sistema permette di rendere un prodotto temporaneamente disponibile o non disponibile.
- Ogni prodotto può essere collegato a uno o più allergeni tramite le funzionalità di gestione.
- Il gestore ha la possibilità di aggiornare e modificare le informazioni relative agli allergeni.
- Il sistema consente la modifica degli ingredienti associati ai singoli prodotti.



ANALISI REQUISITI

REQUISITI FUNZIONALI

- Viene generato automaticamente un codice QR univoco collegato al menù.
- Scansionando il codice QR, il cliente può visualizzare l'elenco dei prodotti presenti nel menù.
- L'accesso al menù da parte del cliente avviene senza necessità di autenticazione.
- Le variazioni apportate al menù vengono mostrate in tempo reale agli utenti che lo consultano.



ANALISI REQUISITI

REQUISITI NON FUNZIONALI

- Il caricamento del menù deve avvenire in modo rapido, riducendo al minimo i tempi di attesa.
- Ogni aggiornamento apportato al menù deve essere immediatamente visibile agli utenti.
- L'interfaccia del sistema deve risultare chiara, facile da usare e comprensibile anche per utenti non esperti.
- Il menù deve adattarsi correttamente alla visualizzazione su smartphone, tablet e computer desktop.
- Il servizio deve garantire un'elevata disponibilità, risultando accessibile in qualsiasi momento della giornata.



OVERVIEW DEL PROBLEMA

Lo scopo del progetto è la realizzazione di un sistema di menù digitale consultabile tramite codice QR, pensato per attività di ristorazione come ristoranti e pizzerie, in alternativa al tradizionale menù su supporto cartaceo. Attraverso la scansione del QR code posizionato sul tavolo, il cliente può accedere immediatamente al menù dal proprio dispositivo mobile, senza dover effettuare alcuna registrazione.

Il menù digitale deve presentarsi in modo ordinato, intuitivo e di facile lettura, così da consentire una consultazione semplice e veloce.

Il sistema include inoltre una sezione dedicata alla gestione interna, riservata al proprietario o al personale autorizzato, dalla quale è possibile intervenire sul menù in totale autonomia. Tramite quest'area è consentito aggiornare i prodotti offerti, variare i prezzi e gestire la disponibilità delle portate. In questo modo il menù risulta sempre allineato alla situazione reale del locale, contribuendo a migliorare l'esperienza del cliente e a rendere il servizio più innovativo ed efficiente.

MOCKUP GRAFICO

LATO GESTORE

conferma elimina

nome

ordine

visibile nel menu

MOCKUP GRAFICO

LATO GESTORE

This mockup shows a form for adding a new item. It consists of several input fields and buttons:

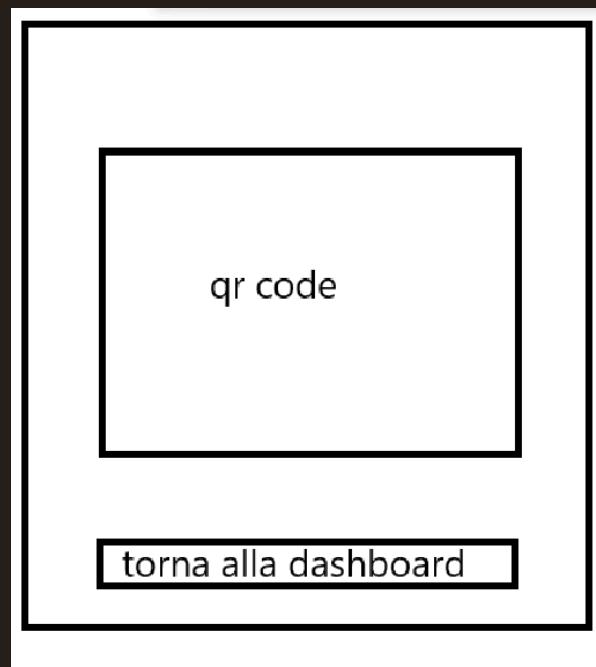
- Text input field labeled "nome" (name)
- Text input field labeled "descrizione" (description)
- Text input field labeled "prezzo" (price) and another labeled "categoria" (category) positioned to its right.
- A checkbox labeled "disponibile" (available) with the label "allergeni" (allergens) positioned below it.
- A checkbox labeled "caratteristiche" (characteristics) with the label "salva" (save) positioned below it.
- A checkbox labeled "annulla" (cancel) positioned next to the "salva" button.

This mockup shows a layout for managing items, featuring the following sections:

- "header" section containing a button labeled "inserisci" (insert).
- "filtr" section containing a button labeled "oggetti presenti" (present objects).
- "nav tab" section.
- "lista elementi" (list of elements) section, which is currently empty.
- "footer" section.

MOCKUP GRAFICO

LATO CLIENTE





OVERVIEW DELLA SOLUZIONE

LINGUAGGIO UTILIZZATO

- HTML
- PHP

API UTILIZZATA

- API QR SERVER

DIAGRAMMA CASI D'USO

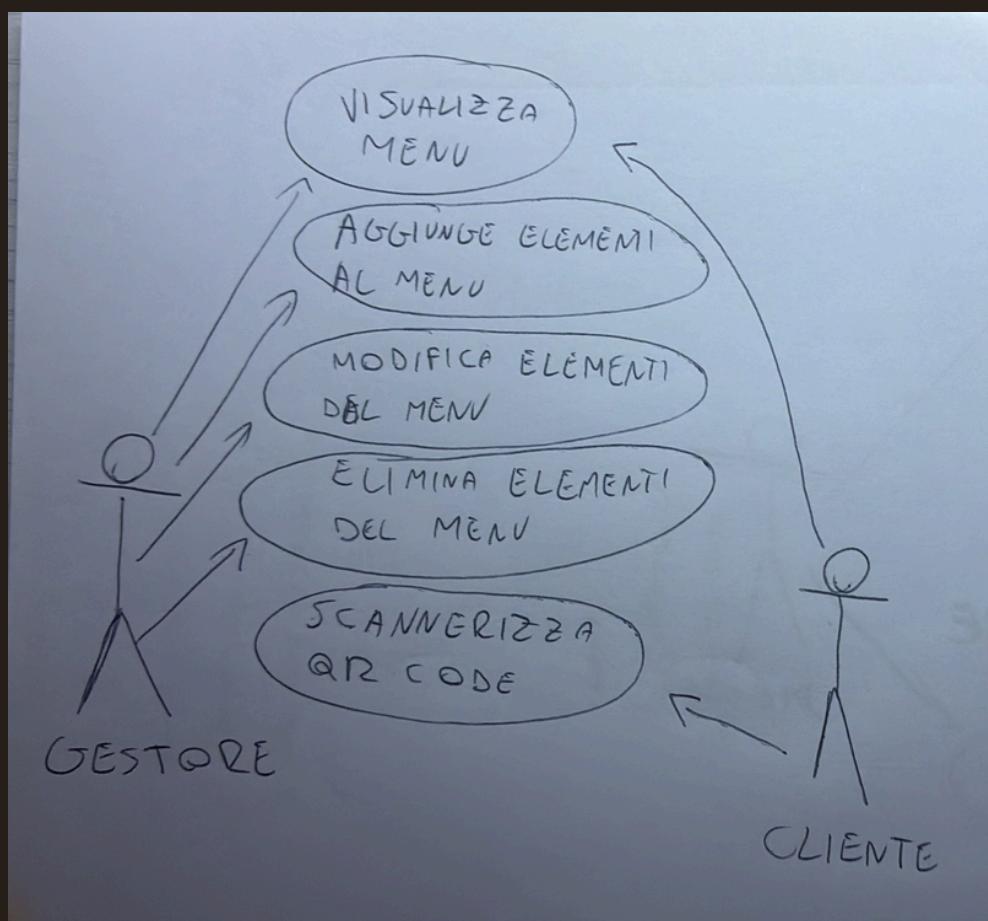
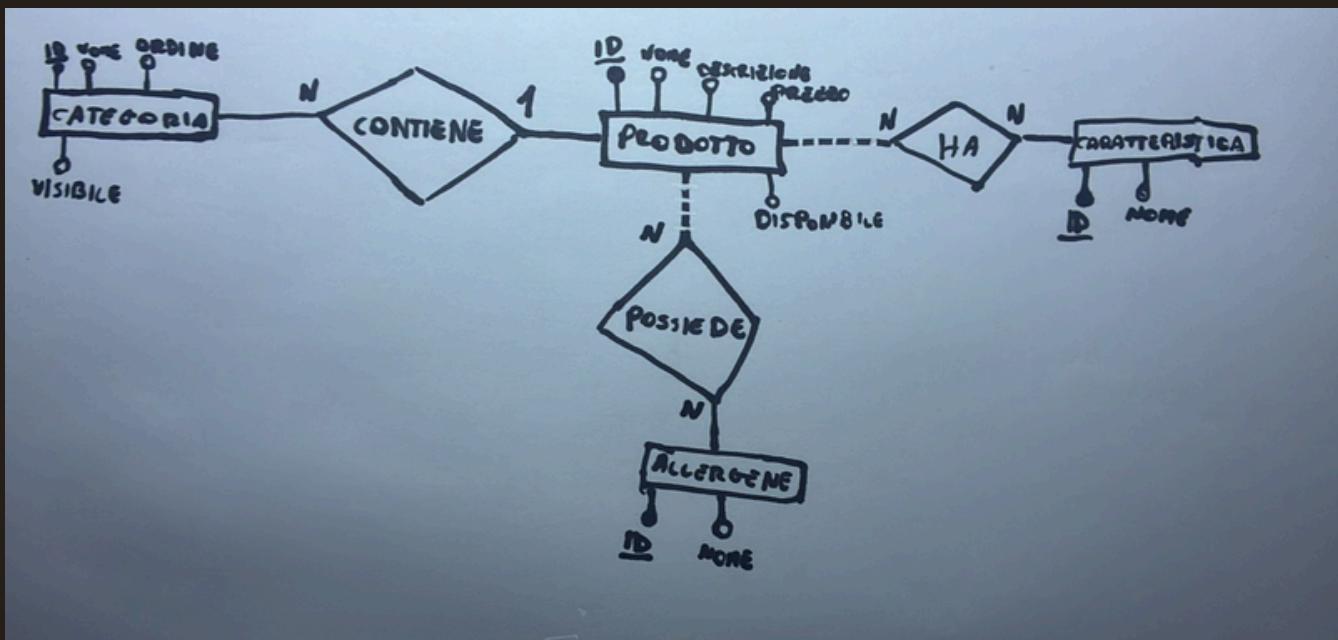


DIAGRAMMA E/R





Progettazione Fisica (Codice SQL DDL)

Di seguito viene riportato il codice SQL utilizzato per la creazione delle tabelle.

-- Tabella Categorie

```
CREATE TABLE `categorie` (
  `id_categoria` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(50) NOT NULL,
  `ordine` int(11) DEFAULT 0,
  `visibile` tinyint(1) DEFAULT 1,
  PRIMARY KEY (`id_categoria`)
) ENGINE=InnoDB;
```

-- Tabella Allergeni

```
CREATE TABLE `allergeni` (
  `id_allergene` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(50) NOT NULL,
  PRIMARY KEY (`id_allergene`)
) ENGINE=InnoDB;
```

-- Tabella Caratteristiche

```
CREATE TABLE `caratteristiche` (
  `id_caratteristica` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(50) NOT NULL,
  PRIMARY KEY (`id_caratteristica`)
) ENGINE=InnoDB;
```

Progettazione Fisica (Codice SQL DDL)

```
-- Tabella Prodotti
CREATE TABLE `prodotti` (
  `id_prodotto` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(100) NOT NULL,
  `descrizione` text DEFAULT NULL,
  `prezzo` decimal(10,2) NOT NULL,
  `disponibile` tinyint(1) DEFAULT 1,
  `immagine` varchar(255) DEFAULT NULL,
  `id_categoria` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_prodotto`),
  CONSTRAINT `fk_prodotti_categorie`
    FOREIGN KEY (`id_categoria`) REFERENCES `categorie` (`id_categoria`)
    ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB;
```

Progettazione Fisica (Codice SQL DDL)

Creazione Tabelle Relazionali

Queste tabelle gestiscono le relazioni Molti-a-Molti (N:M) e utilizzano l'azione ON DELETE CASCADE per garantire che, eliminando un prodotto, vengano rimossi automaticamente anche i suoi legami.

-- Tabella Prodotti-Allergeni

```
CREATE TABLE `prodotti_allergeni` (
  `id_prodotto` int(11) NOT NULL,
  `id_allergene` int(11) NOT NULL,
  PRIMARY KEY (`id_prodotto`, `id_allergene`),
  CONSTRAINT `fk_all_prodotto` FOREIGN KEY (`id_prodotto`) REFERENCES `prodotti`(`id_prodotto`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_all_allergene` FOREIGN KEY (`id_allergene`) REFERENCES `allergeni`(`id_allergene`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```

-- Tabella Prodotti-Caratteristiche

```
CREATE TABLE `prodotti_caratteristiche` (
  `id_prodotto` int(11) NOT NULL,
  `id_caratteristica` int(11) NOT NULL,
  PRIMARY KEY (`id_prodotto`, `id_caratteristica`),
  CONSTRAINT `fk_car_prodotto` FOREIGN KEY (`id_prodotto`) REFERENCES `prodotti`(`id_prodotto`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_car_caratteristica` FOREIGN KEY (`id_caratteristica`)
    REFERENCES `caratteristiche`(`id_caratteristica`) ON DELETE CASCADE ON
    UPDATE CASCADE
) ENGINE=InnoDB;
```