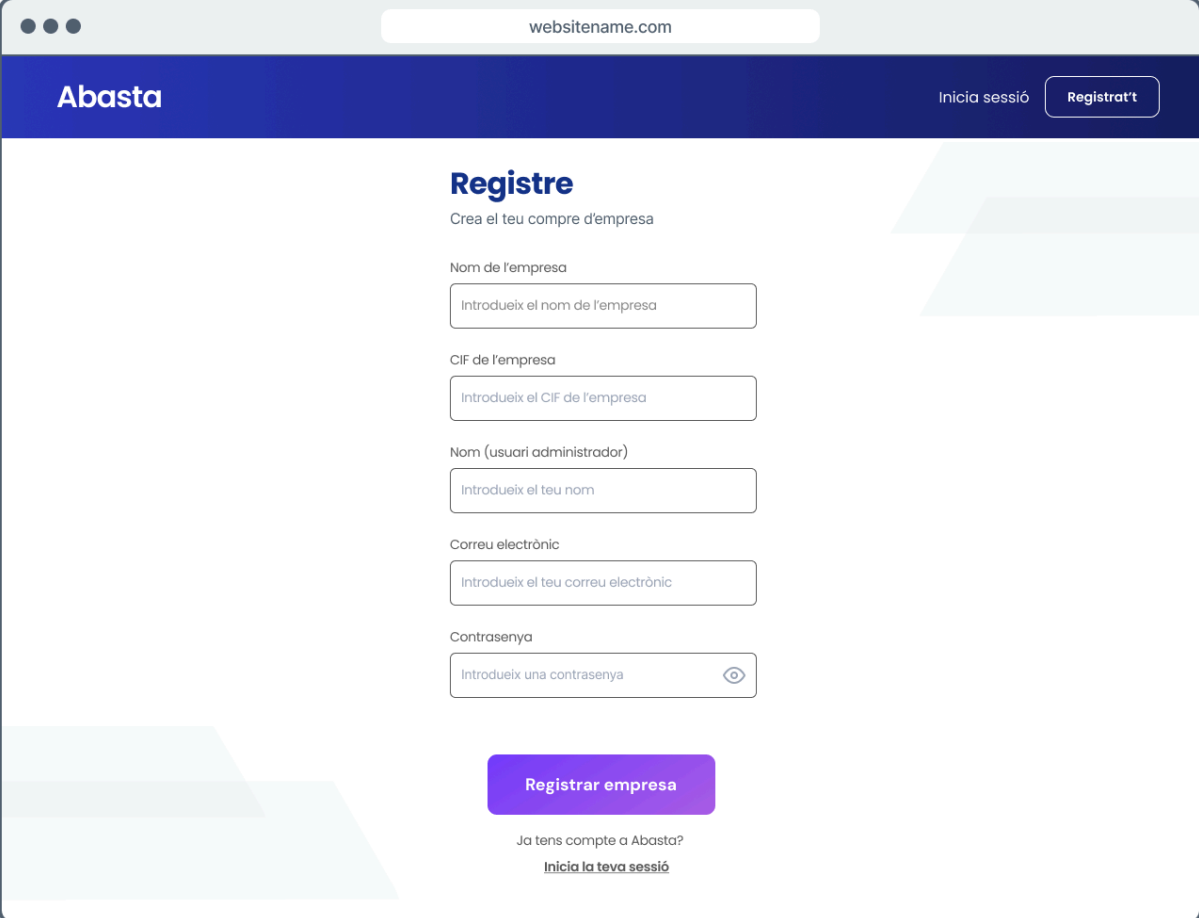


## Documentació del Mòdul de Registre

Aquesta documentació cobreix el component **RegisterForm** (el formulari de registre visible) i el *hook* personalitzat **useRegisterForm** (que gestiona la lògica de l'estat i la comunicació amb el backend).



The screenshot shows a web browser window with the URL 'websitename.com'. The page has a dark blue header with the 'Abasta' logo on the left and 'Inicia sessió' and a 'Registrar't' button on the right. The main content area is white and features a 'Registre' section. Below the title 'Registre' is the subtitle 'Crea el teu compte d'empresa'. The form consists of five input fields: 'Nom de l'empresa' (placeholder: 'Introdueix el nom de l'empresa'), 'CIF de l'empresa' (placeholder: 'Introdueix el CIF de l'empresa'), 'Nom (usuari administrador)' (placeholder: 'Introdueix el teu nom'), 'Correu electrònic' (placeholder: 'Introdueix el teu correu electrònic'), and 'Contrasenya' (placeholder: 'Introdueix una contrasenya' with an eye icon for toggling visibility). A purple 'Registrar empresa' button is positioned below the fields. At the bottom, there is a link that says 'Ja tens compte a Abasta? [Inicia la teva sessió](#)'.

# Abasta

## 1. Component **RegisterForm** (UI)

Aquest component és la interfície d'usuari (UI) que permet als usuaris crear un nou compte d'empresa a la plataforma Abasta.

### Localització

`src/components/auth/RegisterForm/RegisterForm.tsx`

### Funcionalitat

- Mostra un formulari per introduir les dades de l'empresa (Nom, CIF/NIF) i les dades de l'usuari administrador (Nom, Cognoms, Correu, Contrasenya).
- Gestiona la visibilitat de la contrasenya amb una icona (ull).
- Mostra missatges d'error de validació de camp.
- Mostra un missatge d'error general (alerta vermella) si el registre falla.
- Després de l'enviament, substitueix el formulari per una pantalla de notificació d'estat (càrrega i èxit).
- Inclou un botó de tornada a la pàgina principal (**Topbar**).

### Flux i Estats de la Interfície

El component té dos estats de visualització principals, controlats per la variable **isSubmitted**:

| Estat  | Descripció   | Contingut mostrat   |
|--|--|---|
| <b>Formulari Actiu</b><br>( <b>!isSubmitted</b> )  | L'usuari encara no ha enviat el formulari amb èxit.              | <b>Formulari de registre</b> , títols, subtítol i enllaç a <b>Login</b> .             |
| <b>Estat d'Enviament</b><br>( <b>isSubmitted</b> ) | L'usuari ha enviat el formulari i s'està processant la resposta. | Substitueix el formulari per una pantalla de notificació ( <b>register-success</b> ). |

A l'Estat d'Enviament, hi ha dos subestats controlats per **showSuccess**:

1. **Càrrega/Comprovació (!showSuccess)**: Mostra un **Spinner** i el missatge "Comprovant el registre..."
2. **Èxit (showSuccess)**: Mostra una icona de validació (**bi-check-circle**) i el missatge "Compte creat correctament!".

# Abasta

## Camps del Formulari

Tots els camps utilitzen el sistema de registre de `react-hook-form (...register(...))` i mostren errors de validació (`Form.Control.Feedback`) si n'hi ha.

| Camp (ControlId)                | Dades (Hook)                | Validació de Requisits                        |
|---------------------------------|-----------------------------|---|
| <code>formCompanyName</code>    | <code>companyName</code>    | Obligatori                                    |
| <code>formTaxId</code>          | <code>taxId</code>          | Obligatori (CIF / NIF)                        |
| <code>formAdminFirstName</code> | <code>adminFirstName</code> | Obligatori                                    |
| <code>formAdminLastName</code>  | <code>adminLastName</code>  | Obligatori                                    |
| <code>formAdminEmail</code>     | <code>adminEmail</code>     | Obligatori + Patró de correu electrònic vàlid |
| <code>formAdminPassword</code>  | <code>adminPassword</code>  | Obligatori + Mínim <b>8 caràcters</b>         |

# Abasta

## 2. Hook `useRegisterForm` (Lògica)

Aquest *hook* personalitzat aïlla tota la lògica del formulari de registre, la validació, la gestió de l'estat i la comunicació amb el servei d'autenticació.

### Localització

`src/hooks/useRegisterForm.ts`

### Dependències Clau

- **react-hook-form**: Per a la gestió del formulari, la validació i els errors. S'utilitza el mode **onBlur** per a la validació immediata.
- **useAuth**: Per accedir a la funció de registre del context d'autenticació (`registerUser`).

### Estats (Variables React)

| Variable            | Tipus                | Descripció   |
|---------------------|----------------------|--|
| <b>showPassword</b> | <code>boolean</code> | Controla si el camp de contrasenya és de tipus <i>text</i> o <i>password</i> .   |
| <b>isLoading</b>    | <code>boolean</code> | Indica si la petició al servidor està en curs. Deshabilita el botó.  |
| <b>error</b>        | <code>string</code>  | Conté el missatge d'error retornat pel servidor o una excepció genèrica.   |
| <b>isSubmitted</b>  | <code>boolean</code> | Indica que l'enviament de la petició al servidor s'ha realitzat (amb èxit o error inicial).  |
| <b>showSuccess</b>  | <code>boolean</code> | Indica que la resposta del servidor ha estat un èxit, activat 2 segons després de <code>isSubmitted</code> per a simular la comprovació. |

# Abasta

## Funció **onSubmit** (Gestió de l'enviament)

1. Estableix **isLoading** a **true** i esborra qualsevol **error** previ.
2. Executa la funció de registre del context d'autenticació (**registerUser(data)**).
3. **Si té èxit:**
  - Estableix **isSubmitted** a **true** (canviant la UI a la pantalla de notificació).
  - Neteja els camps del formulari (**reset()**).
  - Activa **showSuccess** a **true** després d'un **setTimeout de 2000 ms** per simular un temps de processament o comprovació.
4. **Si hi ha un error (catch):**
  - Registra l'error a la consola.
  - Estableix un missatge d'error genèric (**setError(...)**).
  - **Nota:** En cas d'error, el formulari roman visible (**isSubmitted** és **false** al principi del flux i només es canvia en cas d'èxit, la qual cosa permet a l'usuari corregir les dades).
5. Finalment (**finally**), restableix **isLoading** a **false**.

Aquesta documentació proporciona una base sòlida per entendre com s'implementa i es gestiona el flux de registre a l'aplicació.