

Documentación técnica

Portal de Empleos del Instituto Madero

Claudia Celeste Canta Mayorga

Desarrollo Frontend - Enero 2026

Índice

Descripción General del Proyecto	4
Objetivo General	5
Objetivos Específicos	5
Alcance del Proyecto	5
Tecnologías Utilizadas	6
Frontend	6
Estilos	6
UI y Recursos Visuales	6
Seguridad y Validaciones	6
Backend / Integración	6
Cronograma	6
Exclusiones	6
Flujo Esperado de la Aplicación	7
Inicio de sesión	7
Registro de usuarios	7
Flujo posterior a la autenticación	7
Pantallas	8
Login	8
Propósito	8
Componentes principales utilizados	8
Estado del componente	8
Validaciones	8
Funciones principales	9
Comportamiento de la pantalla	9
Rutas utilizadas	10
Vista de escritorio (Desktop)	10
Vista móvil (Mobile)	11
Selección de Perfil	11
Propósito	11
Componentes principales utilizados	11
Datos de configuración	12
Funciones principales	12
Comportamiento de la pantalla	12
Rutas utilizadas	12
Vista de escritorio (Desktop)	13

Vista móvil (Mobile)	14
Registro de Candidato	14
Propósito	14
Componentes principales utilizados	14
Estado del componente	15
Validaciones	15
Funciones principales	16
Comportamiento de la pantalla	17
Rutas utilizadas	17
Vista de escritorio (Desktop)	18
Vista móvil (Mobile)	18
Registro de Reclutador	19
Propósito	19
Componentes principales utilizados	19
Estado del componente	19
Validaciones	20
Funciones principales	21
Comportamiento de la pantalla	21
Rutas utilizadas	22
Vista de escritorio (Desktop)	22
Vista móvil (Mobile)	23
Home Candidato	23
Propósito	23
Componentes principales utilizados	24
Estado del componente	24
Obtención de datos	25
Filtros y búsquedas	25
Paginación	26
Funciones principales	26
Comportamiento de la pantalla	26
Rutas utilizadas	26
Vista de escritorio (Desktop)	27
Vista móvil (Mobile)	28
Home Reclutador	29
Propósito	29
Componentes principales utilizados	29
Estado del componente	30
Obtención de datos	30

Funciones principales	30
Comportamiento de la pantalla	31
Rutas utilizadas	31
Vista de escritorio (Desktop)	32
Vista móvil (Mobile)	33
Consumo de servicios del backend	34
Propósito general	34
Servicio base de comunicación	35
Manejo centralizado de errores	35
Configuración compartida	35
Cómo ejecutar el proyecto	36
Requisitos previos	36
Instalación del proyecto	36
Configuración del backend	36
Configuración de la API	37
Ejecución del frontend	37
Scripts disponibles	37
Consideraciones importantes	37
Primera ejecución	38

Descripción General del Proyecto

Este proyecto corresponde al desarrollo del frontend del Portal de Empleos del Instituto Madero, con foco en el módulo de acceso a la plataforma. El alcance principal incluye los procesos de registro e inicio de sesión de usuarios, que funcionan como punto de entrada a todas las funcionalidades del sistema.

El sistema permite el ingreso de dos tipos de perfiles:

- **Candidatos**
- **Empleadores**

Cada usuario accede mediante sus propias credenciales y, una vez autenticado, es redirigido a una vista inicial específica según su rol. De esta manera, el sistema adapta la experiencia de navegación de acuerdo con el tipo de usuario que ingresa.

Este módulo constituye la puerta de entrada al portal, siendo un requisito indispensable para el uso de cualquier funcionalidad posterior.

El desarrollo prioriza una experiencia de usuario clara e intuitiva, accesible desde distintos dispositivos, incorporando validaciones en los formularios para asegurar la integridad de los datos ingresados. Asimismo, se implementa una arquitectura frontend modular y mantenible, que sienta las bases para la evolución futura del sistema.

Objetivo General

Desarrollar un módulo frontend de registro y autenticación para el Portal de Empleos del Instituto Madero, que permita el acceso seguro y diferenciado de candidatos y empleadores.

Objetivos Específicos

- Implementar un flujo funcional de registro e inicio de sesión.
- Permitir la creación de perfiles diferenciados según el tipo de usuario.
- Validar los datos ingresados para garantizar un acceso seguro.
- Desarrollar una interfaz responsive, adaptable a distintos dispositivos.

Alcance del Proyecto

El proyecto abarca exclusivamente el desarrollo del frontend encargado de gestionar el acceso de usuarios al portal de empleos, incluyendo:

- Pantallas de registro y login.
- Validaciones de campos y manejo de errores.
- Gestión de estado del formulario y autenticación.
- Redirección a vistas iniciales según el rol del usuario autenticado.
- Integración con una API existente para la validación de credenciales y registro de usuarios.

Las vistas posteriores al inicio de sesión (home de candidato y home de reclutador) cumplen en esta etapa una función demostrativa dentro del flujo de navegación. Si bien presentan opciones y secciones visibles para el usuario, dichas funcionalidades no se encuentran implementadas en la versión actual, quedando previstas para su desarrollo en fases futuras.

Tecnologías Utilizadas

Frontend

- React 18.3.1 – Biblioteca principal para la construcción de la interfaz de usuario.
- TypeScript 5.5.3 – Tipado estático para mejorar la calidad y mantenibilidad del código.
- Vite 5.4.2 – Herramienta de build y servidor de desarrollo.
- React Router DOM 7.10.1 – Gestión de rutas y navegación.

Estilos

- Tailwind CSS 3.4.1 – Framework de utilidades CSS para diseño y responsividad.
- PostCSS 8.4.35 y Autoprefixer 10.4.18 – Procesamiento y compatibilidad de estilos.

UI y Recursos Visuales

- Lucide React 0.344.0 – Biblioteca de íconos utilizada en la interfaz.

Seguridad y Validaciones

- JSEncrypt 3.5.4 – Encriptación RSA de datos sensibles antes de su envío al backend.
- Validator – Validaciones específicas en formularios de registro.

Backend / Integración

- **API REST** – Comunicación con servicios backend configurados en api.config.ts.

Cronograma

Actividad	Descripción	Inicio	Finalización
1. Análisis de requerimientos	Revisión de documentación, mapeo de flujos, identificación de pantallas y requisitos no funcionales	Jun/2025	Jul/2025
2. Diseño de interfaces (UX/UI)	Elaboración de mockups completos de todas las pantallas y validación de diseño	Jul/2025	Sep/2025
3. Desarrollo frontend	Implementación completa del diseño en código, incluyendo componentes, estilos y gestión de estados y lógica	Oct/2025	Dic/2025
4. Integración con backend	Conexión del frontend con la API existente, manejo de respuestas, autenticación y validación del flujo completo	Nov/2025	Dic/2025
5. Pruebas funcionales y ajustes	Pruebas de todas las funcionalidades, validación de formularios y navegación, y corrección de errores detectados	Dic/2025	Dic/2025
6. Documentación	Preparación de la documentación técnica y funcional, y organización del paquete de entrega	Dic/2025	Enero/2026

Exclusiones

- No se realizan desarrollos ni modificaciones en el backend.
- No se implementan nuevas APIs ni lógica del lado del servidor.
- Quedan fuera del alcance otros módulos del portal (publicaciones, búsqueda de empleos, gestión avanzada de perfiles).

Flujo Esperado de la Aplicación

Inicio de sesión

En esta pantalla, el usuario debe ingresar sus credenciales: email y contraseña. La aplicación valida los datos y envía la solicitud de autenticación a la API.

- Autenticación exitosa:
 - Se almacenan los datos del usuario.
 - Se ofrece la opción de recordar el email.
 - Redirección según rol: 'Candidato' o 'Empresa (Reclutador)'
- Error de autenticación: Se muestran mensajes claros indicando credenciales inválidas u otros problemas.

Registro de usuarios

Desde la pantalla de inicio de sesión, el usuario también puede acceder a la opción "Creá tu cuenta", que lo redirige a una pantalla donde debe seleccionar el tipo de perfil con el que desea registrarse.

Registro de Candidato

- Se completa el formulario de registro.
- Se validan los campos ingresados.
- La contraseña se encripta antes de enviarla.
- Los datos se envían a la API.
- Tras un registro exitoso, el usuario es redirigido nuevamente a la pantalla de inicio de sesión.

Registro de Reclutador

- Se completa el formulario de registro.
- Se validan los campos ingresados.
- Los datos se envían a la API.
- Tras un registro exitoso, el usuario vuelve a la pantalla de inicio de sesión.

Flujo posterior a la autenticación

Una vez que el usuario se ha autenticado correctamente:

- Candidato → Pantalla inicial del candidato
- Reclutador → Pantalla inicial del reclutador

Estas pantallas funcionan como vistas base dentro del flujo de navegación y servirán como punto de partida para el desarrollo de futuras funcionalidades.

Pantallas

Login

Propósito

La pantalla Login permite a los usuarios autenticarse en la aplicación. Dependiendo del rol del usuario, se redirige a la pantalla correspondiente. Incluye validación de datos, manejo de errores, opción de recordar el email y navegación hacia registro o recuperación de contraseña.

Componentes principales utilizados

- AuthAside: Sección lateral con contenido visual e informativo.
- PageHeader: Encabezado principal con título y subtítulo.
- FormField: Campo de formulario reutilizable para email y contraseña, con validación y mensajes de error.
- Button: Botón de envío del formulario con estados deshabilitado y carga.
- ErrorMessage: Componente para mostrar errores de login.

Estado del componente

Variable	Tipo	Descripción
email	string	Almacena el correo ingresado por el usuario.
password	string	Almacena la contraseña ingresada por el usuario.

loading	boolean	Indica si la solicitud de login está en proceso.
loginError	string null	Mensaje de error general de login.
emailError	string	Mensaje de error específico del campo email.
passwordError	string	Mensaje de error específico del campo contraseña.
rememberMe	boolean	Controla la opción de recordar el email en localStorage.

Validaciones

- Email
 - Obligatorio.
 - Máximo 50 caracteres.
 - Debe incluir un @.
- Contraseña
 - Obligatoria.
 - Máximo 30 caracteres.

Las funciones validateEmail y validatePassword actualizan los estados de error correspondientes y devuelven un booleano indicando si la entrada es válida.

Funciones principales

- **Gestión del correo electrónico:** Se implementa mediante la función handleEmailChange. Se ejecuta cuando el usuario modifica el campo de correo electrónico, actualizando el estado email con el valor ingresado y validando inmediatamente que cumpla con los criterios de formato y longitud establecidos.
- **Gestión de la contraseña:** La función handlePasswordChange se encarga de actualizar el estado password cada vez que el usuario escribe su contraseña. Además, valida que el valor ingresado no esté vacío y que se encuentre dentro del límite de caracteres permitido.
- **Envío del formulario de inicio de sesión:** El proceso de autenticación es gestionado por la función handleLogin. Al enviarse el formulario, se evita la recarga de la página y se verifica que no exista una solicitud en curso mediante el estado loading. Luego se validan los campos de email y contraseña; si alguno es inválido, el flujo se detiene y se muestran los errores correspondientes.

Cuando los datos son correctos, la función invoca `AuthService.login(email, password)` para autenticar al usuario y posteriormente almacena la información mediante `AuthService.saveUser`. Si el usuario seleccionó la opción Recordar mi usuario, el correo se guarda en `localStorage`; en caso contrario, se elimina cualquier valor almacenado previamente.

Finalmente, el usuario es redirigido según su rol utilizando `useNavigate` de React Router. Al finalizar todo el proceso, el estado `loading` se actualiza para indicar que la operación concluyó.

Comportamiento de la pantalla

Al ingresar a la pantalla, se verifica si existe un correo almacenado en `localStorage`. Si es así, se inicializa el campo `email` con ese valor y se activa la opción 'Recordar mi usuario'.

El usuario puede completar el formulario e interactuar con los campos. Los mensajes de error de `email` y contraseña se muestran únicamente cuando el usuario escribe valores inválidos o intenta enviar el formulario, y aparecen debajo de cada campo correspondiente.

La opción 'Recordar mi usuario' puede activarse o desactivarse, determinando si el correo se guarda para futuras sesiones. La pantalla también incluye enlaces para recuperar la contraseña y para registrarse, que redirige a la selección de perfil.

El botón 'Ingresar' se deshabilita cuando existen errores en los campos o mientras la solicitud está en proceso, evitando envíos duplicados o datos inválidos.

Rutas utilizadas

Ruta	Descripción
<code>/login</code>	Ruta actual de la pantalla Login.
<code>/home-candidato</code>	Redirección tras login exitoso para usuarios tipo Candidato.
<code>/home-reclutador</code>	Redirección tras login exitoso para usuarios tipo Reclutador.
<code>/seleccion-de-perfil</code>	Enlace hacia selección de perfil para registro.

Vista de escritorio (Desktop)

Portal de Empleos Instituto Madero

Iniciar sesión

Conectá con oportunidades y talento.

Correo electrónico

Contraseña

☒ Recordar mi usuario [¿Olvidaste tu contraseña?](#)

Ingresar

¿Todavía no estás registrado?
[Creá tu cuenta como Candidato o Empresa.](#)

Vista móvil (Mobile)

**Portal de Empleos
Instituto Madero**

Iniciar sesión
Conectá con oportunidades y talento.

Correo electrónico
empleador.datasolutions@gmail.com

Contraseña
Ingresá tu contraseña

☒ Recordar mi usuario [¿Olvidaste tu contraseña?](#)

Ingresar

[¿Todavía no estás registrado?
Creá tu cuenta como Candidato o Empresa.](#)

Selección de Perfil

Propósito

La pantalla SeleccionPerfil permite a los usuarios elegir el rol con el que desean registrarse en la aplicación: Candidato o Empresa (Reclutador). Según la elección, el usuario es redirigido al formulario de registro correspondiente. La pantalla también ofrece un enlace para iniciar sesión si ya tiene cuenta.

Componentes principales utilizados

- Header: Encabezado principal de la pantalla, común a toda la aplicación.

- Card & CardContent: Contenedor visual para cada opción de rol, con diseño consistente y responsivo.
- Button: Botón de acción para seleccionar el rol y navegar a la ruta de registro correspondiente.

Datos de configuración

El componente utiliza un arreglo roleCards que define la información de cada tarjeta de rol:

Campo	Descripción
question	Pregunta que describe la función o intención del rol.
buttonText	Texto que se muestra en el botón de selección.
buttonVariant	Estilo del botón: default o outline.
buttonClassName	Clases de estilo aplicadas al botón.
navigateTo	Ruta a la que se redirige al usuario al seleccionar el rol.

Funciones principales

- **Navegación entre pantallas:** Permite cambiar de una pantalla a otra dentro de la aplicación sin recargar la página. La navegación se realiza utilizando el hook useNavigate de React Router.
- **Selección de rol:** Al hacer clic en uno de los botones de rol, el usuario es redirigido al formulario de registro correspondiente según la opción elegida:
 - Candidato (formulario de registro de candidato)
 - Empresa (formulario de registro de reclutador)
- **Acceso a inicio de sesión:** Si el usuario ya tiene una cuenta, puede hacer clic en el enlace 'Inicié sesión', que lo redirige a la pantalla de login.

Comportamiento de la pantalla

Al ingresar a la pantalla, se muestran dos tarjetas con las opciones de perfil disponibles: Candidato y Empresa (Reclutador), cada una acompañada de una breve descripción.

Cuando el usuario selecciona un rol, es redirigido inmediatamente al formulario de registro correspondiente.

La pantalla no realiza validaciones ni consume servicios, ya que su única función es definir el flujo inicial del proceso de registro.

También se incluye un enlace para usuarios que ya poseen una cuenta, permitiendo navegar directamente a la pantalla de inicio de sesión.

Rutas utilizadas

Ruta	Descripción
/seleccion-perfil	Ruta actual de la pantalla Selección de Perfil.
/registro-candidato	Redirección al formulario de registro de Candidato.
/registro-reclutador	Redirección al formulario de registro de Reclutador.
/login	Redirección a la pantalla de Login.

Vista de escritorio (Desktop)

Iniciá sesión'."/>

Portal de Empleos
Instituto Madero

Seleccioná tu rol
Elegí el perfil que mejor te representa.

¿Estás buscando trabajo o querés postularte a nuevas oportunidades?
Candidato

¿Querés publicar búsquedas y encontrar talento para tu equipo?
Empresa

¿Ya tenés cuenta? [Iniciá sesión](#)

Vista móvil (Mobile)

Portal de Empleos
Instituto Madero

Seleccioná tu rol

Elegí el perfil que mejor te representa.

¿Estás buscando trabajo o querés postularte a nuevas oportunidades?

Candidato

¿Querés publicar búsquedas y encontrar talento para tu equipo?

Empresa

¿Ya tenés cuenta? [Inicia sesión](#)

Registro de Candidato

Propósito

La pantalla Registro de Candidato permite a los usuarios crear una cuenta en la aplicación con el rol de candidato. A través de un formulario, el usuario ingresa sus datos personales, credenciales de acceso, un enlace a su currículum y selecciona al menos una habilidad.

Incluye validaciones de campos, manejo de errores, carga dinámica de habilidades desde la API, encriptación de la contraseña antes del envío y redirección a la pantalla de login una vez completado el registro con éxito

Componentes principales utilizados

- HeaderLogo: Logo institucional ubicado en el encabezado de la pantalla.

- Input: Campos de entrada reutilizables para datos personales, email, contraseña y URL del currículum.
- Label: Etiquetas descriptivas para cada campo del formulario.
- Button: Botón de envío del formulario y botones de navegación hacia login o registro de reclutador.
- Select / SelectTrigger / SelectContent / SelectItem: Componentes para la selección de habilidades disponibles.
- ErrorMessage: Componente reutilizable para mostrar errores generales o errores provenientes de la API.
- Iconos Eye / EyeOff: Permiten mostrar u ocultar la contraseña y la confirmación de contraseña.

Estado del componente

Variable	Tipo	Descripción
name	string	Almacena el nombre del candidato.
lastName	string	Almacena el apellido del candidato.
email	string	Almacena el correo electrónico ingresado.
password	string	Almacena la contraseña ingresada.
confirmPassword	string	Almacena la confirmación de la contraseña.
cvLink	string	Almacena el enlace al currículum del candidato.
selectedSkills	string[]	Lista de habilidades seleccionadas.
showPassword	boolean	Controla la visualización de la contraseña.
showConfirmPassword	boolean	Controla la visualización de la confirmación de contraseña.
loading	boolean	Indica si el proceso de registro está en curso.
error	string null	Mensaje de error general del registro.
skillOptions	{ value, label }[]	Lista de habilidades disponibles obtenidas desde la API.
loadingSkills	boolean	Indica si las habilidades se están cargando.

skillsLoadError string | null Error al cargar la lista de habilidades.

Validaciones

- Nombre y Apellido
 - Obligatorios.
 - Máximo 20 caracteres.
- Email
 - Obligatorio.
 - Máximo 60 caracteres.
 - Debe tener formato de correo válido.
- Contraseña
 - Obligatoria.
 - Máximo 30 caracteres.
 - No permite caracteres especiales.
- Confirmación de contraseña
 - Debe coincidir con la contraseña ingresada.
- Currículum (URL)
 - Obligatorio.
 - Debe ser una URL válida con protocolo http o https.
 - Máximo 100 caracteres.
- Habilidades
 - Se debe seleccionar al menos una habilidad.

Si alguna validación falla, el envío del formulario se detiene y se muestran los mensajes de error correspondientes.

Funciones principales

- **Carga de habilidades:** Al inicializar el componente, se ejecuta un efecto que consulta la API para obtener la lista de habilidades disponibles. Estas se formatean y almacenan en el estado skillOptions. Durante este proceso se manejan estados de carga y error para informar al usuario si ocurre algún inconveniente.
- **Gestión de habilidades seleccionadas:** La función handleSkillSelect permite agregar una habilidad a la lista de seleccionadas, evitando duplicados. Y la función handleSkillRemove elimina una habilidad previamente seleccionada, actualizando el estado correspondiente.

- **Gestión del currículum**

La función `handleCvLinkChange` actualiza el estado del enlace al currículum y valida inmediatamente que el valor ingresado sea una URL válida utilizando la librería `validator`.

- **Gestión de contraseña y confirmación**

Las funciones `handlePasswordChange` y `handleConfirmPasswordChange` se encargan de actualizar los estados de contraseña y confirmación, limitando la longitud y evitando caracteres no permitidos.

Además, se controla la visualización de las contraseñas mediante los estados `showPassword` y `showConfirmPassword`.

- **Encriptación de contraseña:** Antes de enviar los datos al backend, la función `encryptPassword` encripta la contraseña utilizando una clave pública RSA obtenida desde las variables de entorno. En caso de error, se envía la contraseña sin encriptar como fallback.
- **Envío del formulario de registro:** El envío del formulario es gestionado por la función `handleSubmit`. Al ejecutarse:
 - Se evita la recarga de la página.
 - Se validan todos los campos del formulario.
 - Si existe algún error, el proceso se detiene y se muestran los mensajes correspondientes.
 - Si los datos son válidos, se construye el objeto de registro y se invoca `CandidateService.registerCandidate`.
 - Al finalizar correctamente, el usuario es redirigido a la pantalla de login.
 - El estado `loading` controla la deshabilitación del formulario durante el proceso.

Comportamiento de la pantalla

Al ingresar a la pantalla, se cargan automáticamente las habilidades disponibles desde la API.

El usuario completa el formulario y recibe feedback inmediato cuando algún campo no cumple con las validaciones.

Las contraseñas pueden mostrarse u ocultarse mediante iconos interactivos.

El botón Registrarse se deshabilita mientras el proceso está en curso para evitar envíos duplicados.

La pantalla incluye accesos rápidos para:

- Iniciar sesión si el usuario ya tiene una cuenta.
- Crear una cuenta como reclutador si corresponde.

Rutas utilizadas

Ruta	Descripción
/registro-candidato	Ruta actual de la pantalla Registro de Candidato.
/login	Redirección tras registro exitoso o acceso manual al login.
/registro-reclutador	Navegación hacia el registro de reclutador.

Vista de escritorio (Desktop)

Portal de Empleos
Instituto Madero

Creá tu cuenta como candidato y accedé a ofertas laborales

Nombre *
Ingresá tu/s nombre/s

Apellido *
Ingresá tu/s apellido/s

Correo electrónico *
Ingresá tu correo electrónico

Contraseña *
Creá una contraseña segura

Repetir contraseña *
Confirmá tu contraseña

Curriculum (URL) *
Link a tu CV en PDF o Drive

Habilidades *
Seleccioná habilidades

HTML ✕ JavaScript ✕ Python ✕

Registrarse

¿Ya tenés cuenta? [Iniciá sesión](#)

¿Sos reclutador? [Crear cuenta](#)

Vista móvil (Mobile)

Portal de Empleos
Instituto Madero

Creá tu cuenta como candidato y accedé a ofertas laborales

Nombre *
Ingresa tu/s nombre/s

Apellido *
Ingresa tu/s apellido/s

Correo electrónico *
Ingresa tu correo electrónico

Contraseña *
Creá una contraseña segura

Repetir contraseña *
Confirmá tu contraseña

Currículum (URL) *
Link a tu CV en PDF o Drive

Habilidades *
Seleccioná habilidades

HTML X JavaScript X Python X

Registrarse

¿Ya tenés cuenta? [Iniciá sesión](#)

¿Sos reclutador? [Crear cuenta](#)

Registro de Reclutador

Propósito

La pantalla Registro de Reclutador permite a los usuarios crear una cuenta con perfil de empresa (reclutador) dentro de la aplicación. A través de un formulario, el usuario ingresa sus datos personales, credenciales de acceso y selecciona la empresa a la que pertenece desde una lista obtenida dinámicamente desde la API.

Incluye validaciones de campos, manejo de errores, carga de empresas desde el backend, encriptación de la contraseña antes del envío y redirección a la pantalla de login una vez finalizado el registro con éxito.

Componentes principales utilizados

- HeaderLogo: Logo institucional mostrado en el encabezado de la pantalla.
- Input: Campos de entrada reutilizables para nombre, apellido, email y contraseñas.
- Label: Etiquetas descriptivas para los campos del formulario.
- Button: Botón de envío del formulario y botones de navegación hacia login o registro de candidato.
- Select / SelectTrigger / SelectContent / SelectItem: Componentes utilizados para la selección de la empresa asociada al reclutador.
- ErrorMessage: Componente reutilizable para mostrar errores generales del proceso de registro.
- Iconos Eye / EyeOff: Permiten mostrar u ocultar la contraseña y su confirmación.

Estado del componente

Variable	Tipo	Descripción
name	string	Almacena el nombre del reclutador.
lastName	string	Almacena el apellido del reclutador.
email	string	Almacena el correo electrónico ingresado.
password	string	Almacena la contraseña ingresada.
confirmPassword	string	Almacena la confirmación de la contraseña.
companyId	string	Identificador de la empresa seleccionada.
showPassword	boolean	Controla la visualización de la contraseña.
showConfirmPassword	boolean	Controla la visualización de la confirmación de contraseña.
nameError	boolean	Indica error en el campo nombre.
lastNameError	boolean	Indica error en el campo apellido.
emailError	boolean	Indica error en el campo email.
passwordFormatError	boolean	Indica error de formato o longitud de contraseña.
passwordMismatchError	boolean	Indica que las contraseñas no coinciden.

companyError	boolean	Indica que no se seleccionó una empresa.
loading	boolean	Indica si el registro está en proceso.
error	string null	Mensaje de error general del registro.
companyOptions	{ value, label }]	Lista de empresas obtenidas desde la API.
loadingCompanies	boolean	Indica si las empresas se están cargando.
companiesLoadError	string null	Error al cargar la lista de empresas.

Validaciones

- Nombre y Apellido
 - Obligatorios.
 - Máximo 20 caracteres.
- Email
 - Obligatorio.
 - Máximo 60 caracteres.
- Contraseña
 - Obligatoria.
 - Máximo 30 caracteres.
 - No permite caracteres especiales.
- Confirmación de contraseña
 - Debe coincidir con la contraseña ingresada.
- Empresa
 - Obligatoria.
 - Debe seleccionarse una opción válida desde la lista obtenida de la API.

Al igual que en la pantalla de 'Registro del Candidato', si alguna validación falla, el envío del formulario se detiene y se muestran los mensajes de error correspondientes debajo de cada campo.

Funciones principales

- **Carga de empresas:** Al montarse el componente, se ejecuta un efecto que consulta la API para obtener el listado de empresas disponibles.

La respuesta se procesa y se transforma en opciones utilizables por el componente Select, almacenándose en el estado `companyOptions`. Durante este proceso se manejan estados de carga y error para informar correctamente al usuario.

- **Gestión de contraseñas:** Las funciones `handlePasswordChange` y `handleConfirmPasswordChange` actualizan los estados de contraseña y confirmación, eliminando caracteres no permitidos y limitando la longitud máxima.

La visualización de las contraseñas se controla mediante los estados `showPassword` y `showConfirmPassword`.

- **Encriptación de contraseña:** Antes de enviar los datos al backend, la función `encryptPassword` encripta la contraseña utilizando una clave pública RSA obtenida desde las variables de entorno.

En caso de que la clave no esté disponible u ocurra un error, se utiliza la contraseña sin encriptar como mecanismo de respaldo.

- **Envío del formulario de registro:** El envío del formulario es gestionado por la función `handleSubmit`. Durante este proceso:
 - Se evita la recarga de la página.
 - Se limpian los errores previos.
 - Se validan todos los campos del formulario.
 - Si existe algún error, el flujo se detiene y se muestran los mensajes correspondientes.
 - Si los datos son válidos, se construye el objeto de registro y se envía a la API mediante una solicitud POST.
 - Si el registro es exitoso, el usuario es redirigido a la pantalla de login.
 - El estado `loading` controla la deshabilitación del formulario durante el proceso.

Comportamiento de la pantalla

Al ingresar a la pantalla, se cargan automáticamente las empresas disponibles desde la API.

El usuario completa el formulario y recibe mensajes de error inmediatos cuando algún campo no cumple con las validaciones.

Las contraseñas pueden mostrarse u ocultarse mediante iconos interactivos.

El botón Registrarse se deshabilita mientras el proceso está en curso para evitar envíos duplicados.

La pantalla ofrece accesos directos para:

- Iniciar sesión si el usuario ya posee una cuenta.
- Crear una cuenta como candidato en caso de haber elegido el perfil incorrecto.

Rutas utilizadas

Ruta	Descripción
/registro-reclutador	Ruta actual de la pantalla Registro de Reclutador.
/login	Redirección tras registro exitoso o acceso manual al login.
/registro-candidato	Navegación hacia el registro de candidato.

Vista de escritorio (Desktop)

Portal de Empleos
Instituto Madero

Creá tu cuenta como empresa y publicá ofertas laborales

Nombre *

Ingresá tu/s nombre/s

Apellido *

Ingresá tu/s apellido/s

Correo electrónico *

Ingresá tu correo electrónico

Contraseña *

Creá una contraseña segura

Repetir contraseña *

Confirmá tu contraseña

Empresa *

Seleccioná tu empresa

Registrarse

¿Ya tenés cuenta? [Iniciá sesión](#)

¿Sos candidato? [Crear cuenta](#)

Vista móvil (Mobile)

Portal de Empleos
Instituto Madero

Creá tu cuenta como empresa y publicá ofertas laborales

Nombre *

Apellido *

Correo electrónico *

Contraseña *

Repetir contraseña *

Empresa *

Registrarse

¿Ya tenés cuenta? [Iniciá sesión](#)

Creá tu cuenta como empresa y publicá ofertas laborales

Nombre *

Apellido *

Correo electrónico *

Contraseña *

Repetir contraseña *

Empresa *

Registrarse

¿Ya tenés cuenta? [Iniciá sesión](#)

¿Sos candidato? [Crear cuenta](#)

Home Candidato

Propósito

La pantalla Home Candidato es el punto principal de inicio para los usuarios con rol candidato. Su objetivo es permitir la exploración y búsqueda de ofertas laborales disponibles, aplicando filtros por empresa, puesto y ubicación, así como búsquedas por texto.

La pantalla ofrece una experiencia responsiva, con menú lateral, filtros adaptados a desktop y mobile, paginación de resultados y acceso a funcionalidades clave como navegación, cierre de sesión y visualización de detalles de cada empleo.

Componentes principales utilizados

- HeaderLogo: Logo institucional presente en la barra de navegación superior.
- Footer: Pie de página común a la aplicación.
- Button: Botones reutilizables para acciones de navegación, filtros, paginación y cierre de sesión.
- InputHomeCandidato: Campo de entrada personalizado para búsqueda por área y ubicación.
- Card / CardContent: Contenedores visuales para mostrar las ofertas laborales.
- Iconos (lucide-react): Iconografía utilizada para navegación, filtros, búsqueda y acciones del usuario.

Estado del componente

Variable	Tipo	Descripción
isMenuOpen	boolean	Controla la visibilidad del menú lateral de navegación.
isFilterOpen	boolean	Controla la visibilidad del panel de filtros en vista mobile.
arealInput	string	Texto ingresado para búsqueda por área o puesto.
locationInput	string	Texto ingresado para búsqueda por ubicación.

showAreaSuggestions	boolean	Controla la visualización de sugerencias de áreas.
showLocationSuggestions	boolean	Controla la visualización de sugerencias de ubicaciones.
expandedSections	Record<string, boolean>	Indica qué secciones de filtros están expandidas.
selectedFilters	object	Almacena los filtros seleccionados por categoría.
jobs	AvailableJob[]	Listado completo de empleos obtenidos desde la API.
loading	boolean	Indica si la carga de empleos está en proceso.
error	string null	Mensaje de error en caso de falla al cargar empleos.
currentPage	number	Página actual de resultados.

Obtención de datos

Al montar el componente, se realiza una solicitud al servicio

AvailableJobsService.getAvailableJobs para obtener la lista de empleos disponibles.

Durante este proceso:

- Se activa el estado loading.
- Si la respuesta es exitosa, los empleos se almacenan en el estado jobs.
- Si ocurre un error o la respuesta es inválida, se muestra un mensaje descriptivo al usuario.
- Al finalizar la solicitud, el estado loading se actualiza para reflejar que el proceso concluyó.

Filtros y búsquedas

La pantalla permite filtrar y buscar empleos mediante distintos mecanismos combinables:

- **Búsqueda por área o puesto:** El campo de búsqueda principal permite ingresar texto libre relacionado con el puesto o descripción del empleo. A medida que el usuario escribe, se muestran sugerencias dinámicas basadas en los datos disponibles.

- **Búsqueda por ubicación:** El segundo campo permite filtrar por ciudad o región, mostrando sugerencias dinámicas según las ubicaciones presentes en las ofertas.
- **Filtros por categorías:** El usuario puede filtrar los resultados por:
 - Empresa
 - Puesto
 - Ubicación

Los filtros se muestran como checkboxes:

- En desktop, se presentan en un panel lateral fijo.
- En mobile, se muestran dentro de un panel deslizable desde la parte inferior.

Cada cambio en los filtros reinicia la paginación y actualiza los resultados visibles.

Paginación

Los resultados se muestran paginados, con un máximo de 8 empleos por página.

El usuario puede:

- Navegar entre páginas mediante botones numerados.
- Avanzar o retroceder usando flechas.
- Al cambiar de página, la vista se desplaza suavemente al inicio del listado.

Funciones principales

- **Gestión de sesión:** La pantalla obtiene la información del usuario autenticado mediante `AuthService.getUser`, mostrando su nombre y rol en el menú lateral.
El cierre de sesión se gestiona a través de `AuthService.logout`, redirigiendo al usuario a la pantalla de login.
- **Filtrado dinámico de empleos:** Los empleos se filtran en memoria teniendo en cuenta:
 - Filtros seleccionados por categoría.
 - Coincidencias con los textos ingresados en área y ubicación. El listado resultante se utiliza para renderizar las tarjetas visibles y calcular la paginación.
- **Visualización de detalle**
Cada tarjeta de empleo incluye la acción “Ver más”, que redirige a la pantalla de detalle del empleo correspondiente mediante React Router.

Comportamiento de la pantalla

Al ingresar a la pantalla:

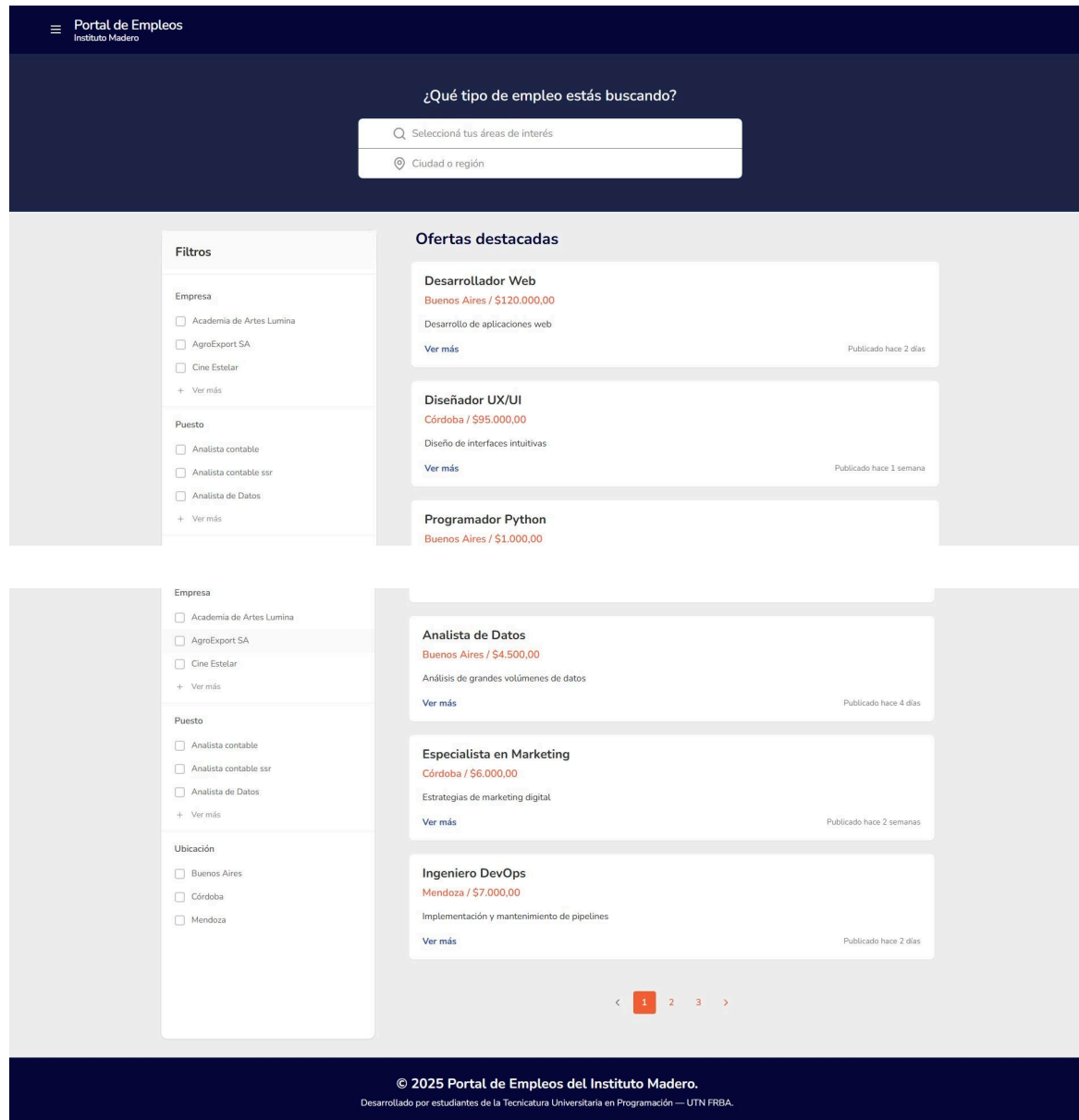
- Se carga automáticamente la lista de empleos.
- Se muestran estados visuales de carga, error o resultados vacíos según corresponda.
- El usuario puede interactuar libremente con filtros, búsquedas y paginación.
- El menú lateral permite acceder a secciones futuras como perfil, postulaciones y configuración.
- La interfaz se adapta automáticamente a desktop y mobile, manteniendo la misma funcionalidad.

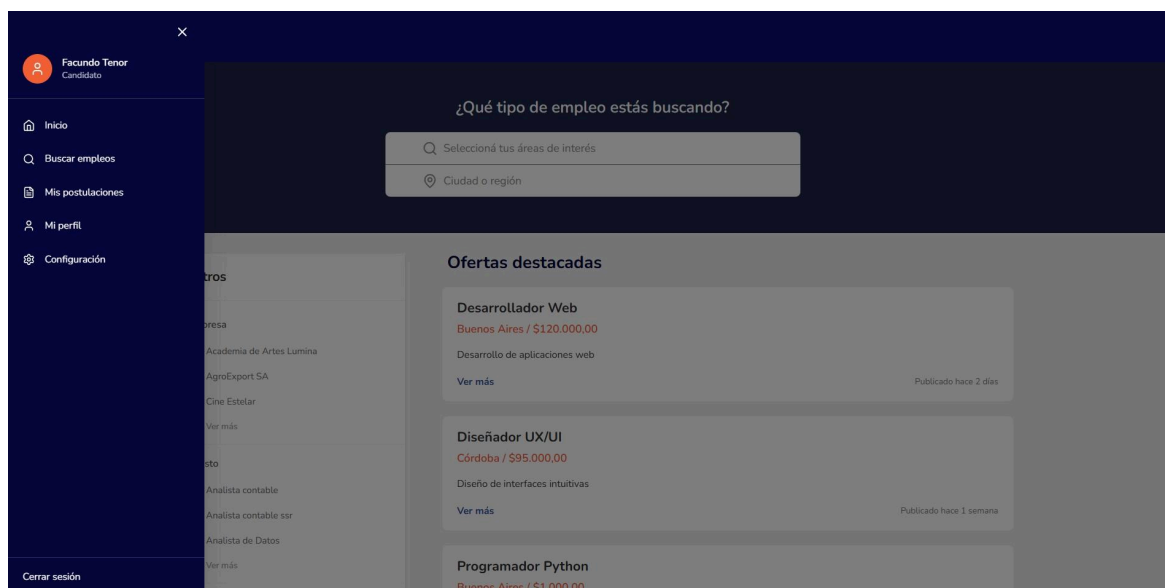
Rutas utilizadas

Ruta	Descripción
/home-candidato	Ruta actual de la pantalla Home Candidato.
/login	Redirección al cerrar sesión.

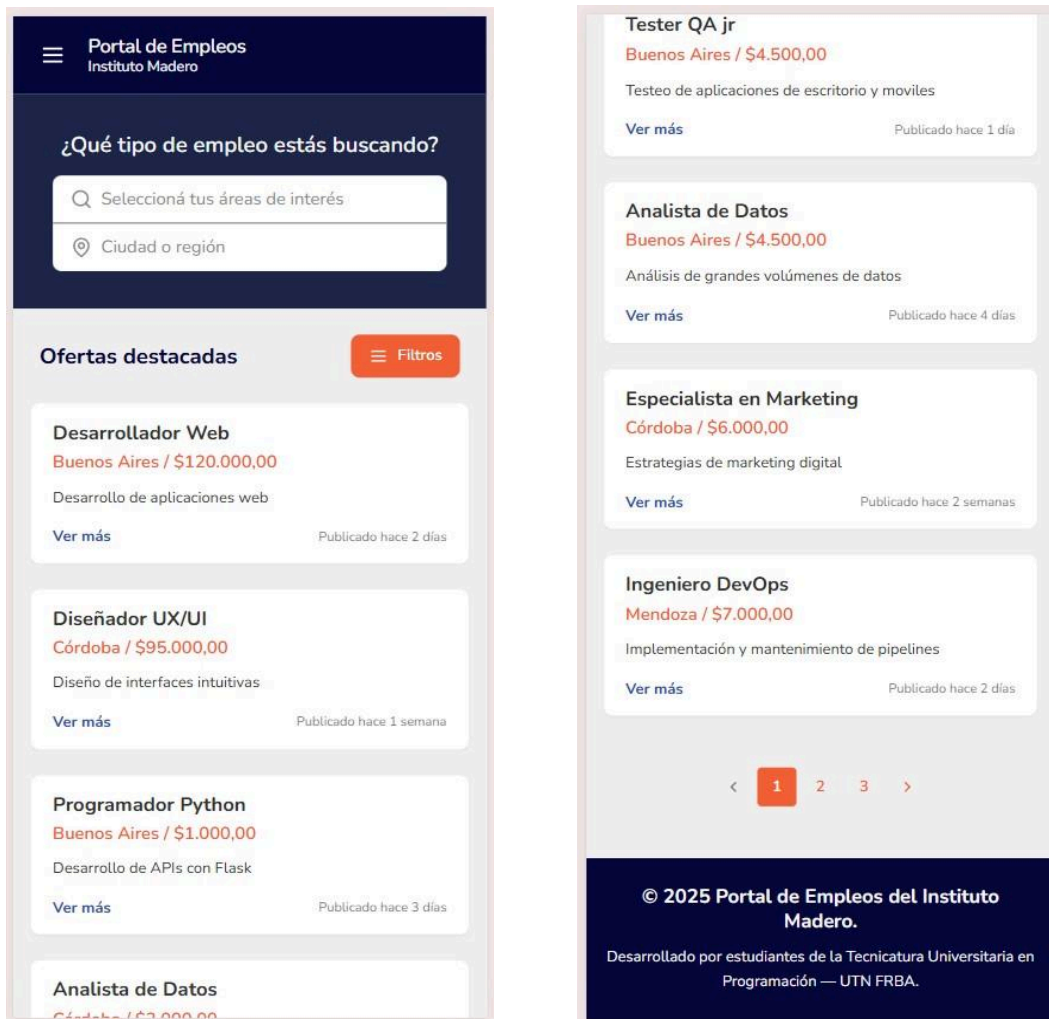
La visualización del detalle de empleo y la postulación directa se encuentran planificadas para una etapa posterior del desarrollo.

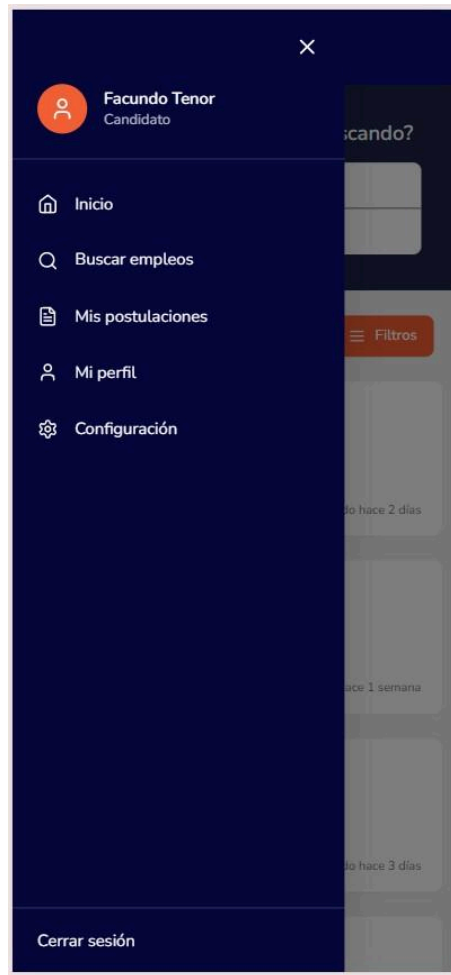
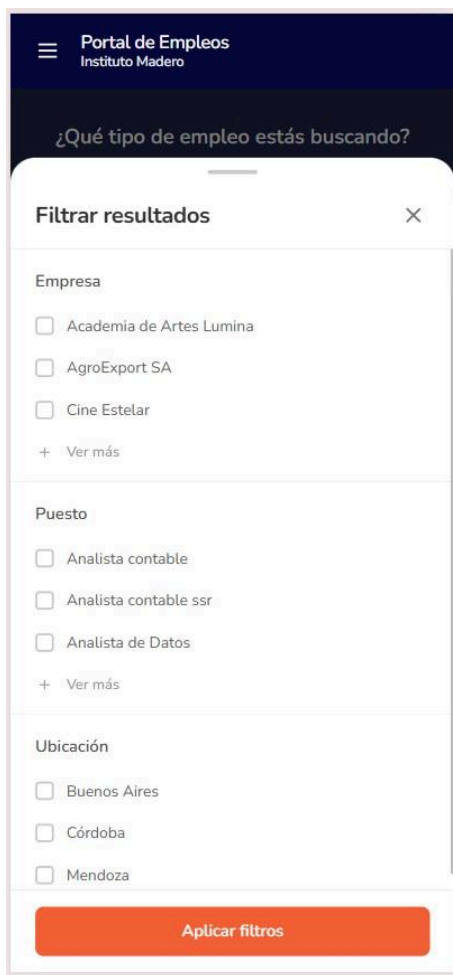
Vista de escritorio (Desktop)





Vista móvil (Mobile)





Home Reclutador

Propósito

La pantalla Home Reclutador funciona como el panel principal para los usuarios con rol reclutador (empresa). Su objetivo es centralizar la gestión de ofertas laborales, permitiendo visualizar publicaciones activas, acceder a secciones administrativas y navegar hacia acciones clave como la creación de nuevas ofertas.

La pantalla actúa como punto de entrada al sistema luego del login exitoso y presenta un resumen del estado general de la cuenta del reclutador.

Componentes principales utilizados

- HeaderLogo: Logo institucional ubicado en la barra superior.

- Button: Botones reutilizables para acciones principales como abrir el menú o crear ofertas.
- SideMenu: Menú lateral con accesos a las secciones principales del sistema y opción de cerrar sesión.
- SearchInput: Campo de búsqueda visual con sugerencias para navegar entre áreas del sistema.
- SectionTitle: Componente reutilizable para títulos de secciones.
- ManagementCard: Tarjetas informativas que resumen el estado de la gestión del reclutador.
- JobCard: Tarjetas que representan las publicaciones laborales recientes.
- Footer: Pie de página común a la aplicación.

Estado del componente

Variable	Tipo	Descripción
isMenuOpen	boolean	Controla la apertura y cierre del menú lateral.
jobs	AvailableJob[]	Listado de ofertas laborales obtenidas desde el servicio.
isLoading	boolean	Indica si las publicaciones se encuentran en proceso de carga.
companyName	string	Nombre de la empresa asociada al reclutador.

Obtención de datos

Al montarse el componente, se ejecuta un efecto que consulta el servicio `availableJobsService.getAvailableJobs`.

Durante este proceso:

- Se activa el estado `isLoading`.
- Se valida el código de respuesta del servicio.
- Se filtran las ofertas correspondientes a la empresa del reclutador (por `company_id`).
- Se actualiza el estado `jobs` con las publicaciones filtradas.
- Se obtiene el nombre de la empresa a partir de los datos recibidos.
- En caso de error, se asignan valores por defecto y se registra el error en consola.
- Finalmente, `isLoading` se actualiza para indicar que la carga finalizó.

Funciones principales

- **Gestión del menú lateral:** El estado `isMenuOpen` controla la visibilidad del menú lateral. El menú se abre mediante el botón de ícono ubicado en la barra superior y se cierra al presionar el botón de cierre o al hacer clic fuera del panel.
- **Gestión de sesión:** La información del usuario autenticado se obtiene mediante `AuthService.getUser`, permitiendo mostrar el nombre y la empresa en el menú lateral. El cierre de sesión se gestiona a través de la función `handleLogout`, que ejecuta `AuthService.logout` y redirige al usuario a la pantalla de login utilizando `useNavigate` de React Router.
- **Búsqueda de secciones (SearchInput):** El componente `SearchInput` permite ingresar texto para filtrar opciones predefinidas de navegación.

Muestra sugerencias dinámicas según el valor ingresado y oculta la lista al perder el foco o hacer clic fuera del componente. Actualmente, la selección de una opción no ejecuta navegación, funcionando únicamente a nivel visual.

- **Visualización de publicaciones recientes:** Las publicaciones laborales se renderizan mediante el componente `JobCard`. Cada tarjeta muestra:
 - Título del puesto y nombre de la empresa.
 - Cantidad de postulaciones.
 - Ubicación.
 - Descripción del puesto.
 - Salario.
 - Fecha de publicación.
 - Botón “Ver detalles”.El botón “Ver detalles” se presenta únicamente a nivel visual, ya que la pantalla de detalle de empleo aún no se encuentra implementada.

Comportamiento de la pantalla

Al ingresar a la pantalla:

- Se muestra un encabezado con el título “Tus ofertas laborales” y una breve descripción.
- Se renderiza el buscador de secciones junto al botón “Crear oferta”.
- Se cargan las tarjetas de gestión general (publicadas, candidatos, cerradas, borradores).
- Se listan las publicaciones recientes asociadas a la empresa.
- Mientras los datos se cargan, se muestra un mensaje de estado.
- Si no existen publicaciones, se informa al usuario.

La interfaz es completamente responsiva y se adapta tanto a vistas de escritorio como móviles.

Rutas utilizadas

Ruta	Descripción
/home-reclutador	Ruta principal del panel del reclutador.
/login	Redirección tras cerrar sesión.

Vista de escritorio (Desktop)

Portal de Empleos
Instituto Madero

Tus ofertas laborales

Visualizá, gestioná y creá nuevas búsquedas laborales.

🔍 Seleccionar área de interés + Crear oferta

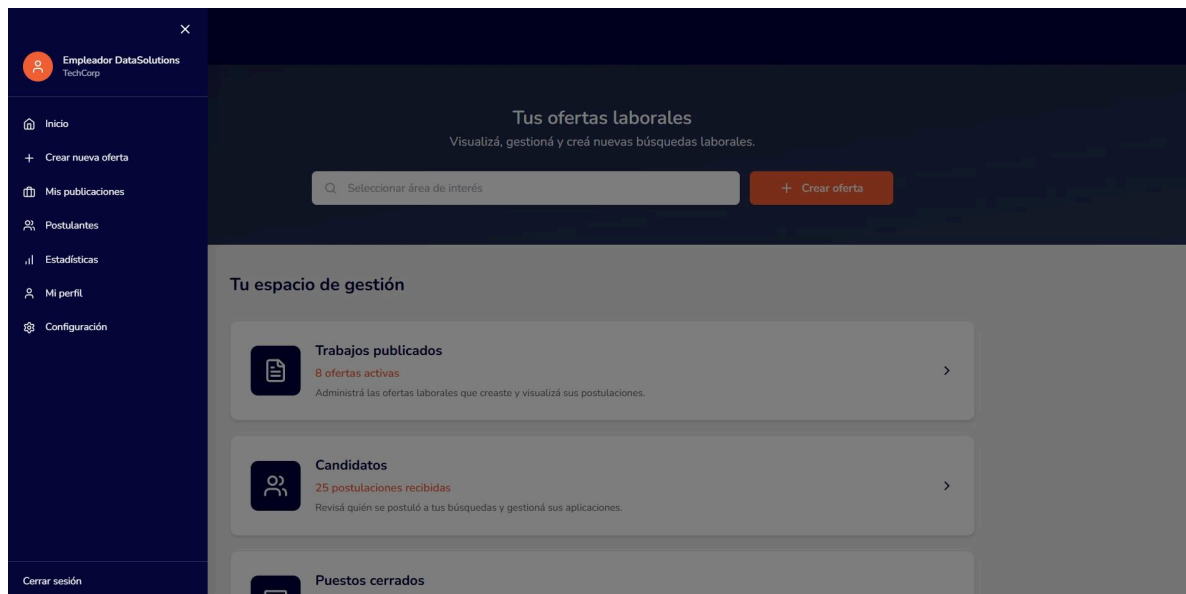
Tu espacio de gestión

- Trabajos publicados**
8 ofertas activas
Administrá las ofertas laborales que creaste y visualizá sus postulaciones.
- Candidatos**
25 postulaciones recibidas
Revisá quién se postuló a tus búsquedas y gestioná sus aplicaciones.
- Puestos cerrados**
Continuá editando ofertas laborales que aún no fueron publicadas.

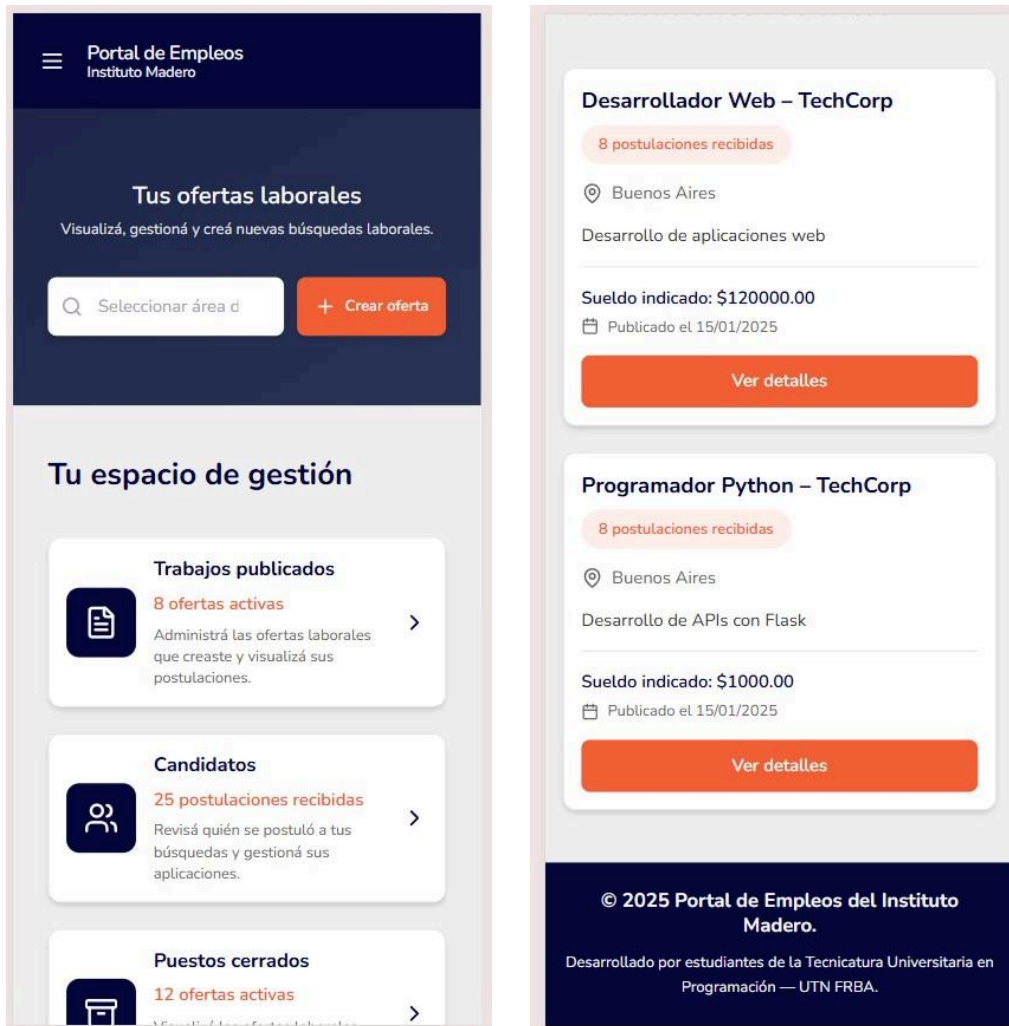
Publicaciones recientes

- Desarrollador Web – TechCorp**
📍 Buenos Aires
Desarrollo de aplicaciones web
Sueldo indicado: \$120000.00
📅 Publicado el 15/01/2025
8 postulaciones recibidas Ver detalles
- Programador Python – TechCorp**
📍 Buenos Aires
Desarrollo de APIs con Flask
Sueldo indicado: \$1000.00
📅 Publicado el 15/01/2025
8 postulaciones recibidas Ver detalles

© 2025 Portal de Empleos del Instituto Madero.
Desarrollado por estudiantes de la Tecnicatura Universitaria en Programación — UTN FRBA.



Vista móvil (Mobile)





Consumo de servicios del backend

Propósito general

Además de las pantallas que conforman la interfaz de usuario, la aplicación interactúa con distintos servicios del backend para poder funcionar correctamente. Estos servicios permiten realizar operaciones como la autenticación de usuarios, el registro, la validación de datos y la obtención de información necesaria para la navegación dentro del sistema.

La comunicación con el backend se realiza a través de servicios centralizados en el frontend, los cuales encapsulan el acceso a los endpoints expuestos por la API y manejan la lógica de consumo, validación de respuestas y manejo de errores.

Servicio base de comunicación

Sobre ApiService se construyen servicios orientados a funcionalidades concretas del sistema:

- **AuthService:** Encargado de la autenticación del usuario y del manejo de la sesión en el frontend. Permite iniciar sesión, persistir los datos del usuario autenticado, validar el estado de autenticación y cerrar sesión.

Endpoint principal utilizado: /portalEmpleos/v1/login

- **CandidateService:** Responsable del registro de usuarios con rol Candidato. Realiza validaciones básicas de los datos antes del envío y maneja los errores de forma centralizada.

Endpoint utilizado: /portalEmpleos/v1/registerCandidateUser

- **EmployerService:** Gestiona el registro de usuarios con rol Empresa (Reclutador), incluyendo validaciones sincrónicas y el manejo estandarizado de errores.

Endpoint utilizado: /portalEmpleos/v1/registerEmployerUser

- **SkillService:** Permite obtener el listado de habilidades disponibles, utilizado principalmente en formularios y filtros.

Endpoint utilizado: /portalEmpleos/v1/getSkillsList

Manejo centralizado de errores

El 'ErrorHandlerService' centraliza el tratamiento de errores provenientes del backend y de problemas de conexión, permitiendo traducir códigos de error en mensajes comprensibles y mantener un comportamiento consistente en toda la aplicación.

Configuración compartida

La aplicación cuenta con una configuración centralizada que define la URL base del backend, los endpoints disponibles y el manejo del token de autenticación.

Este enfoque evita el hardcodeo de valores y facilita cambios de entorno como desarrollo, testing o producción.

Servicio	Endpoint	Pantallas que lo consumen
AuthService	/login	Login
CandidateService	/registerCandidateUser	Registro Candidato

EmployerService	/registerEmployerUser	Registro Reclutador
SkillService	/getSkillsList	Registro Candidato
AvailableJobsService	/getAvailableJobs	Home Candidato, Home Reclutador

Cómo ejecutar el proyecto

A continuación se describen los pasos utilizados para ejecutar el proyecto en un entorno local de desarrollo, de acuerdo con el flujo de trabajo seguido durante el desarrollo del frontend.

Requisitos previos

Para poder ejecutar el proyecto es necesario contar con las siguientes herramientas instaladas:

- Node.js versión 18 o superior
- npm (incluido con Node.js)
- Git (para clonar el repositorio)

Verificación de instalación:

```
node --version
npm --version
git --version
```

Instalación del proyecto

1. Clonar el repositorio.

```
git clone
<https://github.com/cantaclaudia/portal-empleos-frontend>
cd <portal-empleos-frontend>
```

2. Instalar dependencias. Desde la raíz del proyecto ejecutar:

```
npm install
```

Este comando instalará todas las dependencias definidas en el archivo package.json.

Configuración del backend

Antes de ejecutar el frontend, es necesario iniciar el backend y la base de datos correspondiente al proyecto.

1. Abrir Workbench.
2. Ejecutar la base de datos asociada al proyecto, la cual fue provista previamente por el backend.

3. Desde el Explorador de archivos de Windows, ubicarse en la carpeta raíz del proyecto backend.
4. En la barra de direcciones del explorador, escribir powershell y presionar Enter.
5. Ejecutar los siguientes comandos:

```
.\build.bat  
.\run.bat
```

Estos scripts se encargan de construir y levantar el backend junto con sus dependencias necesarias.

Una vez completado este paso, el backend queda disponible para ser consumido por el frontend.

Configuración de la API

El frontend requiere que el backend esté en ejecución para poder funcionar correctamente. La configuración de la API se encuentra centralizada en el archivo 'src/config/api.config.ts'.

Actualmente contiene valores hardcodeados para entorno de desarrollo:

```
BASE_URL: 'http://localhost:5000',  
TOKEN: 'token_test1234'
```

En caso de ser necesario, estos valores pueden modificarse directamente en dicho archivo para apuntar a otro entorno.

Ejecución del frontend

1. Desde el Explorador de archivos, ingresar a la carpeta frontend del proyecto.
2. Hacer clic derecho dentro de la carpeta y seleccionar "Open Git Bash here".
3. Ejecutar el siguiente comando: `npm run dev`

Si la ejecución es correcta, Vite mostrará un mensaje similar al siguiente:

```
VITE v5.x ready in ...  
Local: http://localhost:5173/
```

4. Copiar la URL generada y pegarla en el navegador para visualizar la aplicación.

Scripts disponibles

El proyecto incluye los siguientes scripts definidos en package.json:

```
npm run dev  
npm run build  
npm run preview  
npm run lint  
npm run typecheck
```

Pruebas y validación

Durante el desarrollo del proyecto se realizaron pruebas de integración con el backend para verificar la correcta comunicación entre ambas capas y asegurar la consistencia de los datos consumidos por el frontend.

Las colecciones de Postman utilizadas para estas pruebas fueron provistas por el responsable del desarrollo backend. A partir de ellas, se validó que los endpoints respondieran de manera coherente frente a los mismos datos de entrada, comparando las respuestas obtenidas en Postman con las utilizadas por la aplicación.

Este proceso de validación incluyó no solo los flujos de autenticación (inicio de sesión y registro), sino también la obtención de información necesaria para el funcionamiento del sistema, como los listados de empleos disponibles, habilidades y localidades válidas.

De esta manera, se garantizó que los datos mostrados en la interfaz correspondieran con la información expuesta por la API, permitiendo detectar de forma temprana posibles inconsistencias y asegurando una correcta integración entre frontend y backend.

Consideraciones importantes

- La aplicación se ejecuta en modo desarrollo utilizando Vite.
- El frontend depende del backend y la base de datos, por lo que ambos deben estar ejecutándose previamente.
- La persistencia de sesión se realiza mediante localStorage.
- Las contraseñas se envían encriptadas cuando existe una clave pública RSA configurada.
- El proyecto está optimizado para navegadores modernos (Chrome, Edge, Firefox).
- El enlace generado por `npm run dev` puede variar según la configuración del entorno local.
- El flujo descrito corresponde al entorno utilizado durante el desarrollo del proyecto.

Primera ejecución

En la primera ejecución es normal que:

- `npm install` demore algunos minutos.
- Vite optimice dependencias al iniciar.
- Se muestren mensajes informativos en la consola.