

Ele pega os dados do formulário, valida se estão corretos e cria a conta diretamente no sistema de autenticação do Firebase.

O Firebase cuida de tudo: verifica se o email é único, criptografa a senha com segurança e armazena o usuário de forma protegida.

Se der certo, mostra mensagem de sucesso. Se der erro, avisa o problema específico. É um cadastro rápido e automático sem necessidade de banco de dados adicional.

The screenshot shows the Firebase console interface for a project named "PROJETO FINAL". The left sidebar has a dark theme with navigation links like "Visão geral do projeto", "Authentication" (which is selected and highlighted in blue), "Firestore Database", "Categorias dos produtos", "Criação", "Executar", "Analytics", "Ferramentas de desenvolvimento relacionadas", "Firebase Studio", and pricing information for "Spark" and "Fazer upgrade". The main content area is titled "Authentication" and shows the "Método de login" tab selected. It displays "Provedores de login" with a table showing one provider: "E-mail/senha" with status "ativado" (enabled). There is a blue button labeled "Adicionar novo fornecedor". Below this is a section titled "Avançado" with a sub-section for "Autenticação multifator por SMS". A note says: "Permita que usuários incluam uma camada extra de segurança nas contas deles. Assim que essa autenticação for ativada, integrada e configurada, os usuários poderão fazer login com um processo de duas etapas por SMS." A link "Saiba mais" is provided. A note at the bottom states: "★ A MFA e outros recursos avançados estão disponíveis no Identity Platform, uma solução completa de identidade do Google."

```

// 🔥 SUAS CONFIGURAÇÕES DO FIREBASE
const firebaseConfig = {
    apiKey: "AIzaSyCfDLcFEvWSir9FAhjzwBlq2X75PJwN9c4",
    authDomain: "projeto-final-49d86.firebaseioapp.com",
    projectId: "projeto-final-49d86",
    storageBucket: "projeto-final-49d86.firebaseiostorage.app",
    messagingSenderId: "690462093684",
    appId: "1:690462093684:web:05a9ecc5253af15c06fc21",
    measurementId: "G-H42DE917G4"
};

// Inicializar Firebase
try {
    firebase.initializeApp(firebaseConfig);
    console.log("Firebase inicializado com sucesso!");
} catch (error) {
    console.error("Erro ao inicializar Firebase:", error);
}

const auth = firebase.auth();

```

```

// Função para verificar se Firebase está funcionando
function checkFirebaseConnection() {
    return new Promise((resolve) => {
        const unsubscribe = auth.onAuthStateChanged(user => {
            unsubscribe();
            resolve(true);
        }, error => {
            console.error("Erro na conexão Firebase:", error);
            resolve(false);
        });
    });
}

// Form submission
document.getElementById('registerForm').addEventListener('submit', async function(e) {
    e.preventDefault();

    const name = document.getElementById('name').value.trim();
    const email = document.getElementById('email').value.trim();
    const password = document.getElementById('password').value;
    const confirmPassword = document.getElementById('confirmPassword').value;
    const submitBtn = document.getElementById('submitBtn');

```

```
// Validações
if (!name || !email || !password || !confirmPassword) {
    showMessage('Por favor, preencha todos os campos!', 'error');
    return;
}

if (password !== confirmPassword) {
    showMessage('Senhas não coincidem!', 'error');
    return;
}

if (password.length < 6) {
    showMessage('A senha deve ter pelo menos 6 caracteres!', 'error');
    return;
}

// Desabilitar botão durante o processamento
submitBtn.disabled = true;
submitBtn.textContent = 'Cadastrando...';
showMessage('Processando cadastro...', 'loading');

try {
    // Verificar conexão com Firebase
    const isConnected = await checkFirebaseConnection();
    if (!isConnected) {
        throw new Error('Sem conexão com o servidor. Tente novamente.');
    }

    // 1. Cadastrar no Firebase Authentication
    const userCredential = await auth.createUserWithEmailAndPassword(email, password);
    const user = userCredential.user;

    // 2. Atualizar perfil com nome
    await user.updateProfile({
        displayName: name
    });

    // 3. Enviar email de verificação (opcional)
    // await user.sendEmailVerification();

    //  REMOVIDA A CHAMADA PARA sendToPython
    console.log('Usuário criado com sucesso:', {
        uid: user.uid,
        name: name,
        email: email
    });
}
```