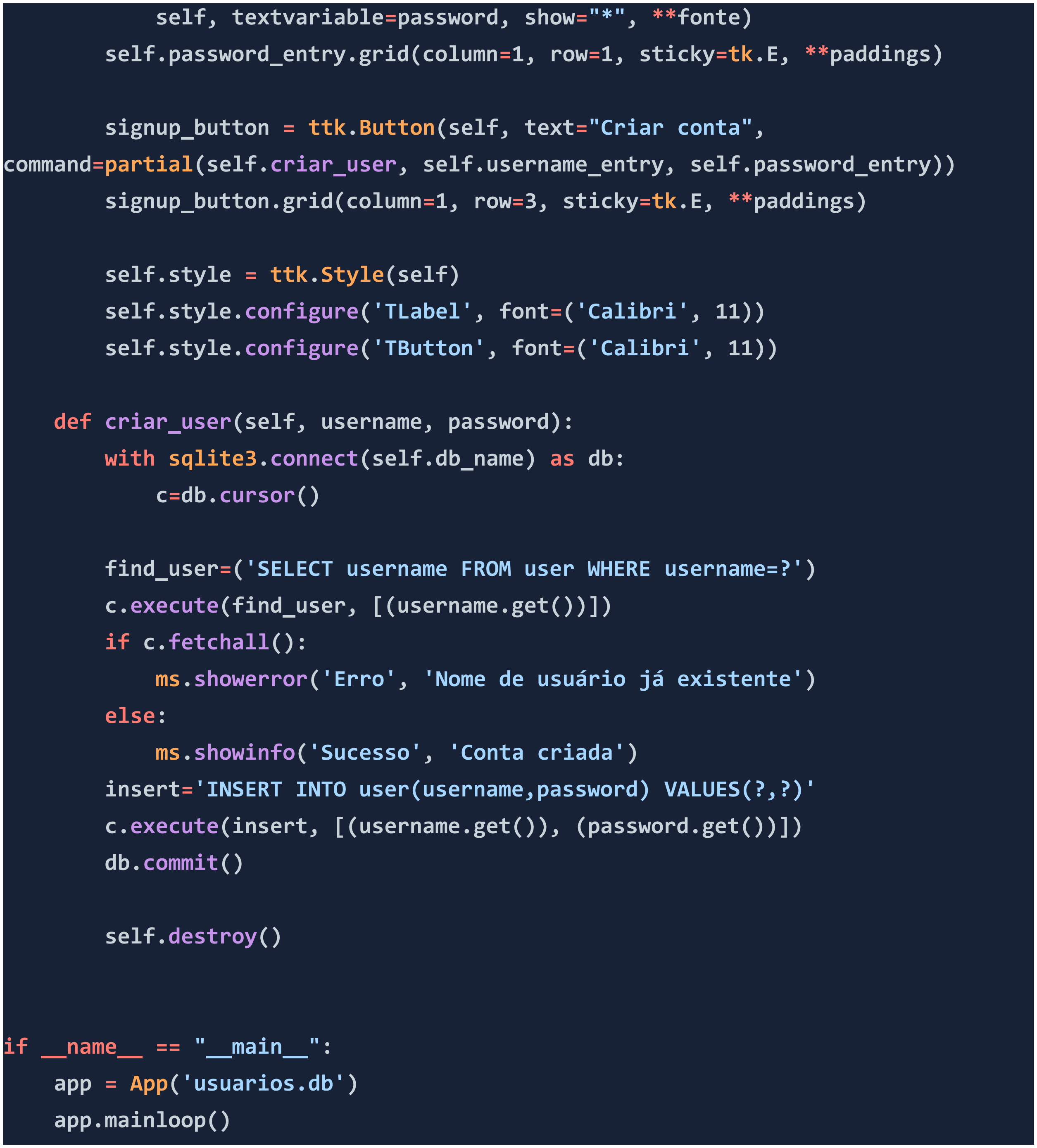
Implementação

```
with sqlite3.connect('usuarios.db') as db:
    c=db.cursor()

c.execute('CREATE TABLE IF NOT EXISTS user (username TEXT NOT NULL PRIMARY KEY,password TEX NOT NULL);')
db.commit()
db.close()

class App(tk.Tk):
    def __init__(self, db_name):
        super().__init__()

        self.geometry('300x110')
        self.resizable(False, False)
        self.eval('tk::PlaceWindow . center')
        self.title('Criar conta')
        self.db_name=db_name

        paddings = {'padx': 5, 'pady': 5}
        fonte = {'font': ('Calibri', 12)}

        self.columnconfigure(0, weight=1)
        self.columnconfigure(1, weight=3)

        username = tk.StringVar()
        password = tk.StringVar()

        username_label = ttk.Label(self, text="Nome de usuário:")
        username_label.grid(column=0, row=0, sticky=tk.W, **paddings)

        self.username_entry = ttk.Entry(self, textvariable=username, **fonte)
        self.username_entry.grid(column=1, row=0, sticky=tk.E, **paddings)

        password_label = ttk.Label(self, text="Senha:")
        password_label.grid(column=0, row=1, sticky=tk.W, **paddings)

        self.password_entry = ttk.Entry(
```



```
            self, textvariable=password, show="*", **fonte)
        self.password_entry.grid(column=1, row=1, sticky=tk.E, **paddings)

        signup_button = ttk.Button(self, text="Criar conta",
command=partial(self.criar_user, self.username_entry, self.password_entry))
        signup_button.grid(column=1, row=3, sticky=tk.E, **paddings)

        self.style = ttk.Style(self)
        self.style.configure('TLabel', font=('Calibri', 11))
        self.style.configure('TButton', font=('Calibri', 11))

    def criar_user(self, username, password):
        with sqlite3.connect(self.db_name) as db:
            c=db.cursor()

        find_user=('SELECT username FROM user WHERE username=?')
        c.execute(find_user, [(username.get())])
        if c.fetchall():
            ms.showerror('Erro', 'Nome de usuário já existente')
        else:
            ms.showinfo('Sucesso', 'Conta criada')
        insert='INSERT INTO user(username,password) VALUES(?,?)'
        c.execute(insert, [(username.get()), (password.get())])
        db.commit()

        self.destroy()


if __name__ == "__main__":
    app = App('usuarios.db')
    app.mainloop()
```

Esse trecho de código representa o requisito funcional 2, onde o usuário se cadastra no aplicativo.