

Banco de inversiones y exchange de Bitfake

Aprendizajes y elementos
más importantes

Índice

1 Elementos más relevantes

2 Dificultades y facilidades

2.1 Aspectos positivos

2.2 Problemas encontrados

2.3 Soluciones

3 Conclusión

Empecemos

1

Elementos más relevantes



1

Realizar transacción
de CLP en cuenta de
banco

2

Ir a la app de bitfake

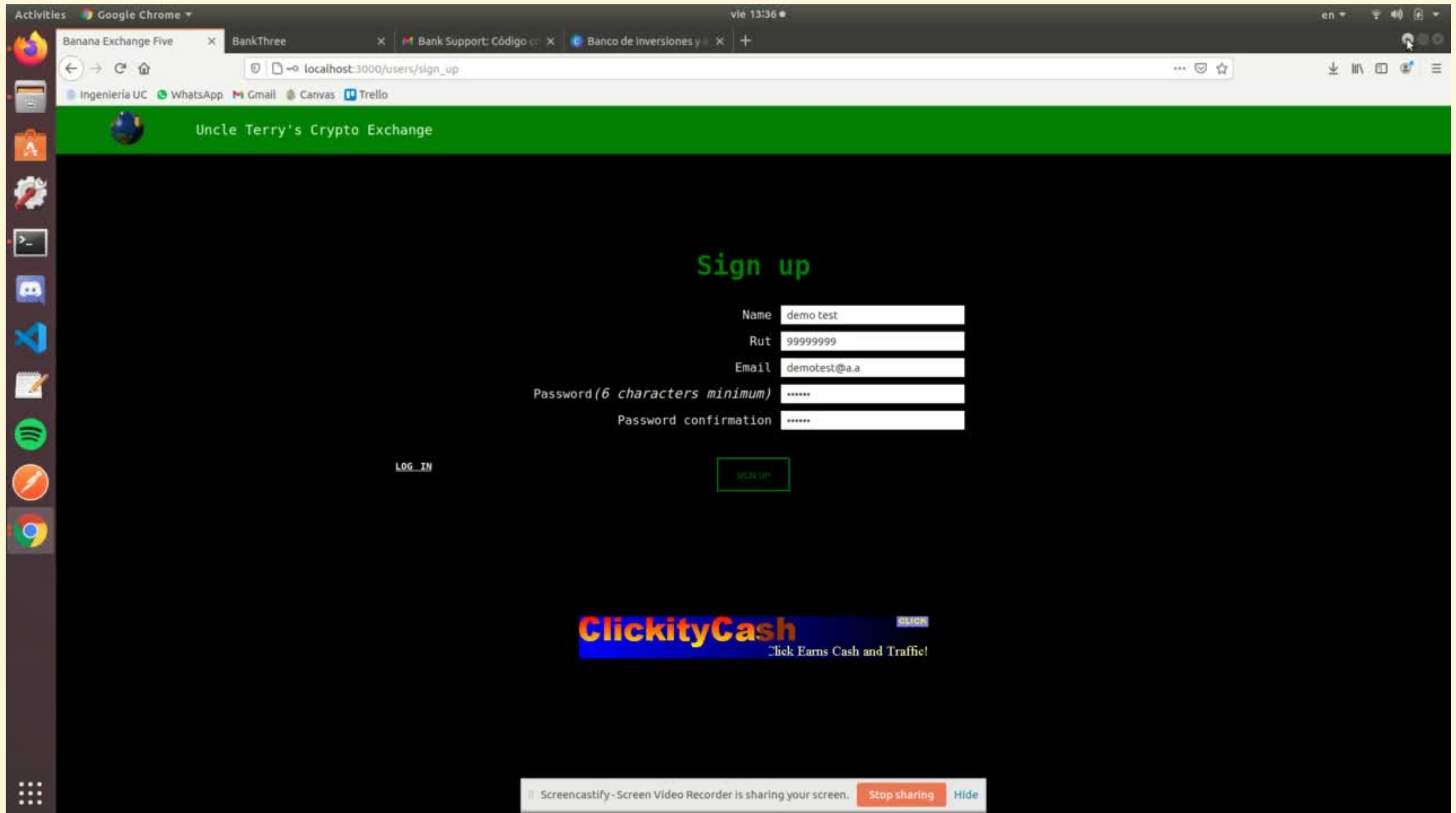
3

Comprar con los CLP
transferidos alguna
moneda.

La conexión con nuestra **API** del banco nos permitió observar y aprender como conectar distintas páginas entre sí, donde en nuestro caso era para realizar transacciones a nuestra cuenta, para luego poder comprar bitcoins en el exchange.

Acompáñenos a ver
una breve demo!





<https://youtu.be/i9nKUBWzKGI>

2

Dificultades y facilidades






POSITIVO

- ✓ **ClassesReferences** → Ayudó a comprender el funcionamiento de la aplicación.
- ✓ **Uso de commands** → El uso de Commands usando la gema "**power-types**" nos facilitó en gran medida la implementación de funcionalidades nuevas:
 - Command **UpdateExchangePrice**
 - Command **ObtainInitialAmount**

PROBLEMAS



-  Buenas prácticas → Algunos nombres de las clases llevaron a malentendidos. Además habían carpetas que confundían el código, ya que no eran utilizadas.
-  Comunicación con la API del banco → Problemas con el *deploy* en Heroku de nuestra aplicación previa.
-  Setup/Build de la aplicación legacy → Errores en los entornos de desarrollo locales de algunos integrantes.

SOLUCIONES



Ingeniería inversa para comprobar el funcionamiento de cada clase




Configurar environment para producción y **permitir autenticación** para el mailer con Gmail app passwords



Aplicación de **Seiketsu (estandarización)**: Todo el equipo adopta WSL 2 como entorno de desarrollo

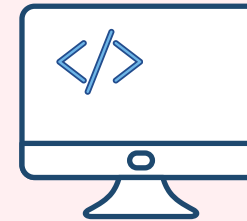
3 Conclusión

Positivo

- Buena documentación. 
- Uso de patrones de diseño, en particular el patrón **Command**.

Problemas

- Importancia de **buenas prácticas** y código **clean**.
- Uso de **entorno** compatible.



Soluciones

- Uso de **ingeniería inversa**.
- Estandarización de WSL 2



Muchas
gracias!

¡Tengan
siempre un
código
CLEAN!

