

The clone (or prototype) pattern

```
// Depending upon your needs, you might not require an abstract
// base class clonable. It would only be needed if you need to store
// clonable objects polymorphically
```

```
class clonable {
public:
    virtual ~clonable() {} // distrutt. polimorfo
    virtual clonable* clone() const = 0; // c.copia polimorfo
};

class Base : public clonable {
public:
    virtual Base* clone() const { return new Base( *this ); }
};

class Derived : public Base {
public:
    virtual Derived* clone() const { return new Derived( *this ); }
};
```

```
void copy_me(const Base& b) {
    Base* clone = b.clone();
    ...
    // delete clone;
};
```